A “Density-Based” Algorithm for Cluster Analysis Using Species Sampling Gaussian Mixture Models

Argiento, R.; Cremaschi, A.; Guglielmi, A.. A “Density-Based” Algorithm for Cluster Analysis Using Species Sampling Gaussian Mixture Models. JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS. 23 pp: 1126-1142.
DOI: 10.1080/10618600.2013.856796

A Bayesian Nonparametric Mixture Model for Cluster Analysis

Raffaele Argiento[1], Andrea Cremaschi[2] and Alessandra Guglielmi[3]
[1]CNR-IMATI, [2]University of Kent and [3]Politecnico di Milano

January 28, 2013

Abstract

In this paper we propose a new model for cluster analysis in a Bayesian nonparametric framework. Our model combines two ingredients, species sampling mixture models of Gaussian distributions on one hand, and a deterministic clustering procedure (DBSCAN) on the other. Here, two observations from the underlying species sampling mixture model share the same cluster if the distance between the densities corresponding to their latent parameters is smaller than a threshold. We complete this definition in order to define an equivalence relation among data labels. The resulting new random partition is coarser than the one induced by the species sampling mixture. Of course, since this procedure depends on the value of the threshold, we suggest a strategy to fix it. In addition, we discuss implementation and applications of the model to a simulated bivariate dataset from a mixture of two densities with a curved cluster, and to a dataset consisting of gene expression profiles measured at different times, known in literature as Yeast cell cycle data. Comparison with more standard clustering algorithm will be given. In both cases, the cluster estimates from our model turn out to be more effective. A primary application of our model is to the case of data from heavy tailed or curved clusters.

Keywords: Bayesian Nonparametrics, Species sampling mixture models, Cluster analysis, DBSCAN.

1 Introduction

In this paper we propose a Bayesian nonparametric model for cluster analysis. Typically, clustering means discovering significant groups (clusters) of data points which belong together because they are similar in some way. Equivalently, the aim is to partition a set of n objects (i.e. data) into k groups, even if the common features of the objects in each group are unknown or unobservable (i.e. latent). In general, the data points do not belong to a unique correct clustering, but, depending on the application, we would like to estimate a "true" one.

There is plenty of cluster analysis algorithms or models that, in the last decades, have been proposed. Here, we find useful to distinguish between model-based and heuristic clustering techniques. The former class refers to those methods that require a statistical model to describe the problem, i.e. mixture modeling; see, for instance, McLachlan and Peel (2000). The latter class includes those algorithms defined from a given starting partition, and carried on following some heuristic scheme. Very popular examples are the hierarchical clustering (Johnson, 1967), and k-means (MacQueen, 1967). While these methods have been widely used in practice, they may suffer from serious limitations. For example, a distance between the objects must always be available, but in general it depends on problem features and data characterization. Moreover, for some of these methods, the number of clusters must be fixed in advance.

Here we propose a Bayesian nonparametric model, that combines two ingredients: species sampling mixture models of Gaussian distributions, and a heuristic clustering procedure, called DBSCAN. The DBSCAN algorithm (Ester et al., 1996) is a density-based clustering technique, where the word *density* refers to the spatial disposition of the data points, that are *dense* when forming a group. DBSCAN requires three input parameters: a distance between data points, the minimum number N of points to define a group to be a cluster, and a threshold representing the maximum distance between elements of the same cluster. Two data points are in the same cluster if their distance is smaller than the threshold; moreover, a cluster is defined using the parameter N (see Ester et al., 1996). As far as the species sampling mixture model is concerned, it is well-known that this model is convenient in order to assign a prior directly on the partition of the data, representing the natural parameter in the cluster analysis context. Moreover, the number of clusters is not fixed a priori, but it is estimated as a feature of the partition of the observations. See Lee et al. (2012) for a recent review on this class of models. However, here, instead of considering the prior on the random partition ρ induced from the species sampling mixture, we consider a deterministic transformation of ρ as a new parameter. The Bayesian cluster estimate will be given in terms of this new random partition, and will result from the minimization of the posterior expectation of a loss function, as usually done in the literature (see Lau and Green, 2007, among the others).

To summarize, our model is based on the slackness of the natural clustering rule of species sampling mixture models of parametric densities, when we mean that two observations X_i and X_j are in the same cluster if, and only if, the latent parameters θ_i and θ_j are equal. We say instead that two observations share the same cluster if the distance between the densities corresponding to their latent parameters is smaller than a threshold ϵ. We complete the definition in order to provide an equivalence relation among data labels. The resulting new

random partition parameter ρ_ϵ is coarser than the original ρ, i.e. the number of elements in ρ_ϵ is smaller than those in ρ. Moreover, under the new parametrization, data within clusters are not independent, and come from a finite mixture of Gaussian densities. Of course, this procedure depends on the value of the threshold ϵ and the distance between densities. As far as the latter choice is concerned, we use Hellinger distance, symmetrized Kullback-Leibler I-divergence and L^2 distance, since they are easy to interpret, and have a closed analytical form under Gaussian kernels. On the other hand, the elicitation of a value for ϵ has a key role in our model, since this threshold greatly affects the cluster estimate. Here we suggest to fix a grid of reasonable values for ϵ, and choose the value maximizing the posterior expectation of a function of the random partition. In the applications, we used some predictive distribution summary statistics, as well as more standard tools like the silhouette coefficient and the adjusted Rand index.

In this work, we have decided to focus on Gaussian kernels, but of course other parametric families could be fixed as well. The choice of the Gaussian distribution is essentially due to nice theoretical properties (mixtures of Gaussians are dense in the space of densities on an Euclidean space), low computational effort (conjugacy), and closed form of some distances (L^2, Kullback-Leibler and Hellinger).

We discuss implementation and applications of the model to a simulated bivariate dataset from a mixture of two densities with a curved cluster, and to a dataset consisting of gene expression profiles measured at 9 different times, known in literature as Yeast cell cycle data. Comparison with more standard clustering algorithms will be given. In both cases, the cluster estimates from our model turn out to be more effective. Our estimates fit data particularly well when they come from heavy tailed or curved clusters.

The rest of this paper is organized as follows. Section 2 describes the underlying species sampling mixture models. In Section 3 we describe the model under the new parametrization in details, discussing some of its main features. A short discussion on the computation is provided in Section 4. Section 5 illustrates the choice of the threshold parameter ϵ. In Section 6 the simulated bivariate "curved" dataset is analyzed, while Section 7 addresses the Yeast cell cycle data in Cho et al. (1998). We conclude with a discussion in Section 8.

2 The model

We set up a Bayesian model in which the partition of data is a random variable, distributed according to some prior distribution. If $(X_1, ..., X_n)$ represents the data, its conditional

distribution is:

$$(X_1, ..., X_n)|C_1, \ldots, C_k, \phi_1, \ldots, \phi_k \sim \prod_{j=1}^{k} \left\{ \prod_{i \in C_j} f(x_i; \phi_j) \right\}, \tag{1}$$

where $\boldsymbol{\rho} := \{C_1, \ldots, C_k\}$ is a partition of the the data label set $\{1, \ldots, n\}$ and $\{f(\cdot; \phi), \phi \in \Theta\}$ is a family of densities on $\mathbb{R}^p$. We require the family of densities to be identifiable, that is, $P_{\phi_1} = P_{\phi_2}$ implies $\phi_1 = \phi_2$, where P_ϕ is the probability measure corresponding to the density $f(\cdot; \phi)$. Observe that here k is the number of clusters in the partition $\boldsymbol{\rho}$. From (1), it is clear that, conditionally on $\boldsymbol{\rho}$, the data are independent between different clusters and are independent and identically distributed (i.i.d.) with density $f(\cdot, \phi)$ within each cluster. To complete the Bayesian model we need to assign a prior for $(\boldsymbol{\rho}, \boldsymbol{\phi})$. As far as $\boldsymbol{\rho}$ is concerned, we will assume that

$$\pi(\boldsymbol{\rho}) = \mathbb{P}(\boldsymbol{\rho} = \{C_1, \ldots, C_k\}) = p(\#C_1, \ldots, \#C_k), \tag{2}$$

where $p(\cdot)$ is an *infinite* exchangeable partition probability function (eppf), i.e. a symmetric function such that $p(1) = 1$ and

$$p(n_1, \ldots, n_k) = \sum_{j=1}^{k} p(\ldots, n_j + 1, \ldots) + p(n_1, \ldots, n_k, 1);$$

see Pitman (1996). Moreover, conditionally on $\boldsymbol{\rho}$, we assume that the parameters in $\boldsymbol{\phi} := (\phi_1, \ldots, \phi_k)$ in (1) are i.i.d. from some fixed distribution P_0 on $\Theta \subset \mathbb{R}^s$. By Pitman (1996), for each distribution P_0 and eppf $p(\cdot)$, there exists a unique species sampling prior $\Pi(\cdot; p, P_0)$ on the space of all probabilities on Θ, such that model (1) under the specified prior is equivalent to

$$\begin{aligned} &X_i|\theta_i \overset{\text{ind}}{\sim} f(\cdot|\theta_i) \quad i = 1, \ldots, n \\ &\theta_i|P \overset{\text{iid}}{\sim} P \quad i = 1, \ldots, n \\ &P \sim \Pi(\cdot; p, P_0), \end{aligned} \tag{3}$$

where P_0 represents the expectation of P. In this model every X_i has density $f(\cdot, \theta_i)$, which is univocally determined by the value of θ_i. In this case, we say that θ_i is the latent variable corresponding to X_i in the mixture model (3).

In this work we will consider only proper species sampling models, that is

$$P(\cdot) = \sum_{i=1}^{\infty} \xi_i \delta_{\boldsymbol{\tau}_i}(\cdot), \text{ where } \begin{cases} (\xi_i) \sim \pi(\cdot; p) \\ (\boldsymbol{\tau}_i) \overset{\text{iid}}{\sim} P_0(\cdot) \end{cases},$$

and (ξ_i) and $(\boldsymbol{\tau}_i)$ are independent. An interesting example is the *Normalized Generalized Gamma* (NGG) process prior, introduced by Regazzini et al. (2003). It is well known that

such a process P can be represented as

$$P = \sum_{i=1}^{+\infty} \xi_i \delta_{\tau_i} = \sum_{i=1}^{+\infty} \frac{J_i}{T} \delta_{\tau_i}$$

where $\xi_i := \frac{J_i}{T}$, $(J_i)_i$ are the ranked points of a Poisson process on $\mathbb{R}$ with mean intensity $\rho(ds)$, and $T = \sum_i J_i$. We write $P \sim NGG(\sigma, \alpha, P_0)$, with parameters (σ, α, P_0), where $0 \leq \sigma \leq 1, \alpha \geq 0$. See Lijoi et al. (2007) and Argiento et al. (2010) for more details. This class encompasses the Dirichlet processes: when $\sigma = 0$ and $\alpha > 0$, P is the Dirichlet process (Ferguson, 1973) with measure parameter $\alpha P_0(\cdot)$.

The eppf $p(\cdot)$ corresponding to a proper species sampling P can be recovered from the following formula:

$$p(n_1, \ldots, n_k) = \sum_{(j_1, \ldots, j_k)} \mathbb{E} \prod_{i=1}^{k} w_{j_i}^{n_i}, \tag{4}$$

were $(j_1, \ldots, j_k)$ ranges over all permutations of k positive integers. See Lijoi et al. (2007) for an explicit expression of $p(n_1, \ldots, n_k)$ under a NGG process prior. On the other hand, when the NGG process prior reduces to the Dirichlet measure, formula (4) turns out to be a variant of Ewens sampling formula (Ewens, 1972):

$$p(n_1, \ldots, n_k) = \frac{\Gamma(\alpha + 1)}{\Gamma(\alpha + n)} \alpha^{k-1} \prod_{j=1}^{k} (n_j - 1)! \ \ ;$$

see also Antoniak (1974).

Hierarchical mixture models as (3) are frequently adopted in the Bayesian nonparametric framework for their mathematical tractability; moreover, the corresponding posterior computations are relatively easy, due to the availability of straightforward MCMC schemes. We will exploit this representation in order to compute posterior inference. On the other hand, formulation (1)-(2) is the most expressive here, since the random parameter contains $\boldsymbol{\rho}$, which is the object of our statistical analysis.

Finally, observe that equivalence between models (1)-(2) on one hand and (3) on the other holds thanks to the natural clustering rule and identifiability of the likelihood. By *natural clustering rule* we mean the following: given $\theta_1, \ldots, \theta_n$, X_i and X_j belong to the same cluster if, and only if, $\theta_i = \theta_j$. In this case we write $X_i \leftrightarrow X_j$. The partition $\boldsymbol{\rho} = \{C_1, \ldots, C_k\}$ represents the quotient set of the equivalence relation on the data label set $\{1, \ldots, n\}$ induced by $\leftrightarrow$, and $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_k)$ are the unique values among the θ_i's.

3 Relaxing the equality constraint in the mixture model

The sensitivity of cluster estimates to hyperparameters in species sampling mixture models is a well-known issue. First of all, when the tails of the "true" distribution are heavy, in order to fit the data, the Bayesian estimate will adopt many kernels to reconstruct the "true" density shape. This occurs in particular when dealing with kernel densities having positive support, such as Weibull or Gamma densities. In this case, a deeper analysis on the prior elicitation could be accomplished. See for instance Argiento et al. (2012) and Griffin (2010). Secondly, if the "true" distribution has non-convex contour lines, as in Section 6 here, the hierarchical mixture model generally will give cluster estimates where the cluster components do not represent real data clusters, unless the kernel density has a proper non-convex shape.

To overcome these problems, here we propose a new rule to assign observations to clusters, relaxing the equality constraint imposed by the natural clustering rule under the species-sampling mixture model (3). If $d(\cdot, \cdot)$ is any distance between densities, the natural clustering rule can be restated as

$$X_i \leftrightarrow X_j \quad \Leftrightarrow \quad d(f(\cdot, \theta_i), f(\cdot, \theta_j)) = 0$$

when the family $\{f(\cdot; \theta), \theta \in \Theta\}$ is identifiable. To relax the rule, instead of grouping elements whose kernel densities are equal, we assign those data whose densities are "close" to the same cluster.

Definition 1. *Given a configuration $(\theta_1, \ldots, \theta_n)$, a threshold $\epsilon > 0$, and a distance between densities $d(\cdot, \cdot)$, two observation X_i and X_j are* directly reachable *if*

$$d(f(\cdot, \theta_i), f(\cdot, \theta_j)) < \epsilon.$$

We write $X_i \overset{\epsilon}{\rightsquigarrow} X_j$; since transitivity does not hold in this case, $\overset{\epsilon}{\rightsquigarrow}$ is not an equivalence relation.

Definition 2. *Given a configuration $(\theta_1, \ldots, \theta_n)$, a threshold $\epsilon > 0$, and a distance between densities $d(\cdot, \cdot)$, two observations are* reachable *if there exist a finite sequence $X_{j_1}, \ldots, X_{j_m}$ such that*

$$X_i \overset{\epsilon}{\rightsquigarrow} X_{j_1} \overset{\epsilon}{\rightsquigarrow} X_{j_2} \overset{\epsilon}{\rightsquigarrow} \ldots \overset{\epsilon}{\rightsquigarrow} X_{j_m} \overset{\epsilon}{\rightsquigarrow} X_j.$$

We write $X_i \overset{\epsilon}{\leftrightarrow} X_j$. It is not difficult to prove that $\overset{\epsilon}{\leftrightarrow}$ is an equivalence relation among the data (see the proof in the Appendix).

It is worthy to note that Definition 1 was given to relax the condition under which two observations are in the same cluster. However, as just observed, $\rightsquigarrow$ is not an equivalence relation and for this reason it does not lead to a partition on the data index set $\{1, \ldots, n\}$. Consequently, Definition 2 was introduced in order to define an equivalence relation. Now

we are able to define $\boldsymbol{\rho}_\epsilon = \{C_1^{(\epsilon)}, \ldots, C_m^{(\epsilon)}\}$ as the quotient set of the equivalence relation $\overset{\epsilon}{\leftrightarrow}$ on $\{1, \ldots, n\}$. Here $m := m(\epsilon) \leq k$ denotes the new number of clusters.

These definitions were suggested by Ester, Kriegel, and Xu (1996), introducing the DBSCAN algorithm (*density-based spatial clustering of applications with noise*). DBSCAN is a well known algorithm in the data mining community; in brief, it clusters data at hand through a notion of distance between items and two parameters, an integer N and a positive real ϵ. In this work we consider only the case $N = 1$, since if $N > 1$, the relation $\overset{\epsilon}{\leftrightarrow}$ induced by DBSCAN among the data labels is not an equivalence. We refer to the original paper Ester, Kriegel, and Xu (1996) for the meaning of N.

In this paper, by DBSCAN$(\{f(\cdot;\theta_1), \ldots, f(\cdot;\theta_n)\}, d, \epsilon)$ we mean a deterministic function: the input values are: (i) $(\theta_1, \ldots, \theta_n)$, the latent variables in model (3) corresponding to the data, (ii) a distance d between densities, (iii) a threshold $\epsilon > 0$, having fixed the kernel density $f(\cdot;\theta)$. The input values $(\theta_1, \ldots, \theta_n)$ can be equivalently described as $(\{C_1, \ldots, C_k\}, (\phi_1, \ldots, \phi_k))$, as it is usually done under DPM models for example, while (ii) can be substituted by a matrix of the distances between $f(\cdot;\phi_i)$ and $f(\cdot;\phi_j)$, $i, j = 1, \ldots, k$. The output values are: (i) a partition $(C_1^{(\epsilon)}, \ldots, C_m^{(\epsilon)})$ of the index set $\{1, \ldots, n\}$, obtained grouping the subset $\{C_1, \ldots, C_k\}$ according the deterministic procedure DBSCAN (i.e. according the equivalence relation $\overset{\epsilon}{\leftrightarrow}$ given by Definitions 1 and 2), and the vectors $(\boldsymbol{\phi}_1^{(\epsilon)}, \ldots, \boldsymbol{\phi}_m^{(\epsilon)})$ (of the latent variables associated to each $C_j^{(\epsilon)}$, $j = 1, \ldots, m$) and $(\boldsymbol{n}_1^{(\epsilon)}, \ldots, \boldsymbol{n}_m^{(\epsilon)})$ (size vectors of the sets among $\{C_1, \ldots, C_k\}$ composing $C_1^{(\epsilon)}, \ldots, C_m^{(\epsilon)}$). Specifically, we are applying the deterministic DBSCAN procedure to the species-sampling mixture model, obtaining a new model, called b-DBSCAN. Let us see what we mean in more details.

The b-DBSCAN model

Applying the DBSCAN procedure to the partition $\boldsymbol{\rho} = \{C_1, \ldots, C_k\}$ with latent variables $(\theta_1, \ldots, \theta_n)$ and unique values $\boldsymbol{\phi} := (\phi_1, \ldots, \phi_k)$ from a species-sampling process, we obtain a new partition $\boldsymbol{\rho}_\epsilon = \{C_1^{(\epsilon)}, \ldots, C_m^{(\epsilon)}\}$ such that, for each $h = 1, \ldots, m$, we have:

$$C_h^{(\epsilon)} = C_{l_1^h} \cup \cdots \cup C_{l_{k_h^\epsilon}^h}, \qquad \{l_1^h, \ldots, l_{k_h^\epsilon}^h\} \subseteq \{1, \ldots, k\}. \tag{5}$$

In brief, (5) states that an element $C_h^{(\epsilon)}$ of the partition $\boldsymbol{\rho}_\epsilon$ is finite union of some elements of the partition $\boldsymbol{\rho}$, which depend on ϵ and the index h. Let us consider now, for each $h = 1, \ldots, m$, the vector $\boldsymbol{\phi}_h^{(\epsilon)} := (\phi_{l_1^h}, \ldots, \phi_{l_{k_\epsilon^h}^h})$ and the vector $\boldsymbol{n}_h^{(\epsilon)} := (\#C_{l_1^h}, \ldots, \#C_{l_{k_h^\epsilon}^h}) = (n_{l_1^h}, \ldots, n_{l_{k_\epsilon^h}^h})$. In the following lines, we will explicit the model w.r.t. the new parameters $\boldsymbol{\rho}_\epsilon$, $\boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)}$, $h = 1, \ldots, m$.

For each $\epsilon > 0$, reordering the multiplication factors and using notation in (5), we can

re-write model (1) as

$$
\begin{aligned}
& X_1, ..., X_n | C_1, \ldots, C_k, \phi_1, \ldots, \phi_k \sim \prod_{h=1}^{m} \prod_{j=1}^{k_h^\epsilon} \Big\{ \prod_{i \in C_{l_j^h}} f(x_i; \phi_{l_j^h}) \Big\} \\
& \phi_1, \ldots, \phi_k | k \overset{\text{iid}}{\sim} P_0 \\
& \boldsymbol{\rho} \sim \pi(\boldsymbol{\rho}) = \text{eppf}(\#C_1, \ldots, \#C_k).
\end{aligned} \tag{6}
$$

From the first line of (6), we see that, conditionally on $\boldsymbol{\rho}$ and $\boldsymbol{\phi}$, the data are independent between the m cluster $C_1^{(\epsilon)}, \ldots, C_m^{(\epsilon)}$. They are also conditionally independent within these clusters; in fact, if $\boldsymbol{X}_{C_h^{(\epsilon)}}$ represents the subvector of data in cluster $C_h^{(\epsilon)}$, then conditionally on $\boldsymbol{\phi}_h^{(\epsilon)}$ and $C_{l_1^h}, \ldots, C_{l_{k_h^\epsilon}^h}$, its density is

$$
\mathcal{L}(\boldsymbol{X}_{C_h^{(\epsilon)}} | \boldsymbol{\phi}_h^{(\epsilon)}, C_{l_1^h}, \ldots, C_{l_{k_h^\epsilon}^h}) = \prod_{j=1}^{k_h^\epsilon} \Big\{ \prod_{i \in C_{l_j^h}} f(\cdot; \phi_{l_j^h}) \Big\}. \tag{7}
$$

This expression tells us that every data component in $\boldsymbol{X}_{C_h^{(\epsilon)}}$ has (conditional) distribution that is a finite mixture, whose kernels are the k_h^ϵ densities $f(\cdot; \phi_{l_1^h}), \ldots, f(\cdot, \phi_{l_{k_h^\epsilon}^h})$. However, since the process generating the groups of labels $C_{l_1^h}, \ldots, C_{l_{k_h^\epsilon}^h}$ is a species sampling scheme, the components of the subvector $\boldsymbol{X}_{C_h^{(\epsilon)}}$ are not independent and identically distributed. To clarify, let $m_h^{(\epsilon)} := \#C_h^{(\epsilon)}$ (it is worth noting that $m_h^{(\epsilon)} = n_{l_1^h} + \cdots + n_{l_{k_\epsilon^h}^h}$), and let us denote by $Z_1^h, \ldots, Z_{m_h^{(\epsilon)}}^h$ the data in $\boldsymbol{X}_{C_h^{(\epsilon)}}$. Moreover, let $\{\eta_1^h, \ldots, \eta_{m_h^{(\epsilon)}}^h\}$ (with values in $\{l_1^h, \ldots, l_{k_\epsilon^h}^h\}$) be the latent variables of $Z_1^h, \ldots, Z_{m_h^{(\epsilon)}}^h$, representing the component in the mixture they are generated from. We have

$$
\mathcal{L}(\boldsymbol{X}_{C_h^{(\epsilon)}} | \boldsymbol{\phi}_h^{(\epsilon)}, \eta_1^h, \ldots, \eta_{m_h^{(\epsilon)}}^h) = \prod_{i=1}^{m_h^{(\epsilon)}} f(z_i^h; \phi_{\eta_i^h}). \tag{8}
$$

The labels $(\eta_1^h, \ldots, \eta_{m_h^{(\epsilon)}}^h)$ yield a partition of the vector $\boldsymbol{X}_{C_h^{(\epsilon)}}$ into subgroups: the components in $\boldsymbol{X}_{C_h^{(\epsilon)}}$ are in the same subgroup if the corresponding labels are equal. In particular, if this subpartition and $\{C_{l_1^h}, \ldots, C_{k_h^\epsilon}^h\}$ coincide, then expression (8) and (7) are equal. Equivalently, the law in (7) is the distribution of $(Z_1, \ldots, Z_{m_h^{(\epsilon)}})$, conditionally to the event that $(\eta_1^h, \ldots, \eta_{m_h^{(\epsilon)}}^h)$ describes the subpartition $\{C_{l_1^h}, \ldots, C_{l_{k_h^\epsilon}^h}\}$.

Now, let us consider the distribution of $(\eta_1^h, \ldots, \eta_{m_h^{(\epsilon)}}^h)$, conditionally to $\boldsymbol{\phi}_h^{(\epsilon)}$ and $\boldsymbol{n}_h^{(\epsilon)}$; from the conditional independence of the data, this latter distribution is the law of a sample without replacement of $m_h^{(\epsilon)}$ elements from the set containing $n_{l_1^h}$ times the number l_1^h, $n_{l_2^h}$ times the number l_2^h, and so on, up to the set containing $n_{l_{k_\epsilon^h}^h}$ times the number $l_{k_\epsilon^h}^h$. Then,

marginally, each η_j^h, $j = 1, \dots, m_h^{(\epsilon)}$, has distribution

$$\mathbb{P}(\eta_j^h = \cdot | \boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)}) = \frac{n_{l_1^h}}{n_{l_1^h} + \cdots + n_{l_{k_\epsilon^h}^h}} \delta_{l_1^h}(\cdot) + \cdots + \frac{n_{l_{k_\epsilon^h}^h}}{n_{l_1^h} + \cdots + n_{l_{k_\epsilon^h}^h}} \delta_{l_{k_\epsilon^h}^h}(\cdot). \tag{9}$$

Now let $\tilde{f}(; \boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)})$ be the density of $\boldsymbol{X}_{C_h^{(\epsilon)}}$ obtained integrating out over the values of the labels $\eta_1^h, \dots, \eta_{m_h^{(\epsilon)}}^h$. By (9), $\tilde{f}(\cdot; \boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)})$ is such that each component Z_j^h, $j = 1, \dots, m_h^{(\epsilon)}$, of $\boldsymbol{X}_{C_h^{(\epsilon)}}$, has distribution with density

$$\frac{1}{dz}\mathbb{P}(Z_j^h = dz | \boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)}) = \frac{n_{l_1^h}}{n_{l_1^h} + \cdots + n_{l_{k_\epsilon^h}^h}} f(z; \phi_{l_1^h}) + \cdots + \frac{n_{l_{k_\epsilon^h}^h}}{n_{l_1^h} + \cdots + n_{l_{k_\epsilon^h}^h}} f(z; \phi_{l_{k_\epsilon^h}^h}). \tag{10}$$

Summing up, now we are able to re-write the model as follows:

$$\begin{aligned}
&X_1, \dots, X_n | C_1^{(\epsilon)}, \dots, C_m^{(\epsilon)}, \boldsymbol{\phi}_1^{(\epsilon)}, \dots, \boldsymbol{\phi}_m^{(\epsilon)}, \boldsymbol{n}_1^{(\epsilon)}, \dots, \boldsymbol{n}_m^{(\epsilon)} \sim \prod_{h=1}^{m} \tilde{f}(\boldsymbol{X}_{C_h^{(\epsilon)}}; \boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)}) \\
&(C_1^{(\epsilon)}, \dots, C_m^{(\epsilon)}, \boldsymbol{\phi}_1^{(\epsilon)}, \dots, \boldsymbol{\phi}_m^{(\epsilon)}, \boldsymbol{n}_1^{(\epsilon)}, \dots, \boldsymbol{n}_m^{(\epsilon)}) = \text{DBSCAN}(\{C_1, \dots, C_k\}, (\phi_1, \dots, \phi_k), d, \epsilon) \\
&\phi_1, \dots, \phi_k | k \overset{\text{iid}}{\sim} P_0 \\
&\boldsymbol{\rho} \sim \pi(\boldsymbol{\rho}) = \text{eppf}(\#C_1, \dots, \#C_k),
\end{aligned} \tag{11}$$

where the density $\tilde{f}(\cdot; \boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)})$ has been described above. We will refer to (11) as b-DBSCAN model in the rest of the paper. In conclusion, we elicit the prior on the parameter of interest $(\boldsymbol{\rho}_\epsilon, \boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)}, h = 1, \dots, m) := \text{DBSCAN}(\{C_1, \dots, C_k\}, (\phi_1, \dots, \phi_k), d, \epsilon)$, as the prior induced by a deterministic transformation of $(\boldsymbol{\rho}, \boldsymbol{\phi})$.

To make inference, we will need to sample from the posterior distribution

$$\mathcal{L}(\boldsymbol{\rho}_\epsilon, \boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)}, h = 1, \dots, m | data). \tag{12}$$

Observe that, augmenting the state space, (12) is the marginal distribution of

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\rho}_\epsilon, \boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)}, h = 1, \dots, m, \boldsymbol{\rho}, \boldsymbol{\phi} | data) = \\
&\qquad \mathcal{L}(\boldsymbol{\rho}_\epsilon, \boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)}, h = 1, \dots, m | \boldsymbol{\rho}, \boldsymbol{\phi}, data) \mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\phi} | data).
\end{aligned} \tag{13}$$

Since $\{\boldsymbol{\rho}_\epsilon, \boldsymbol{\phi}_h^{(\epsilon)}, \boldsymbol{n}_h^{(\epsilon)}, h = 1, \dots, m\}$ is a deterministic function of $(\boldsymbol{\rho}, \boldsymbol{\phi})$, the first factor on the right hand-side of (13) is degenerate on $\text{DBSCAN}(\{C_1, \dots, C_k\}, (\phi_1, \dots, \phi_k), d, \epsilon)$. The second factor is the posterior distribution of the parameter $(\boldsymbol{\rho}, \boldsymbol{\phi})$ in model (1)-(2).

In the rest of the paper we will fix ϵ, without assuming it random. In fact, from (6), it is clear that, conditionally to $(\boldsymbol{\rho}, \boldsymbol{\phi})$, the distribution of the data does not depend on ϵ. This also implies that, as far as density estimation is concerned, models (3) and (11) are equivalent.

4 Computational details

One of the main advantages of the DBSCAN procedure is that it is very fast ($O(n \log n)$), and very simple to implement. Moreover, the package "fpc" (Hennig, 2012) of the R software (R Development Core Team, 2012) contains a function implementing DBSCAN algorithm, given a distance matrix among the data.

Now, when considering model (11) for a fixed ϵ, all the cluster inferences are based on $\mathcal{L}(\boldsymbol{\rho}_\epsilon|data)$. To obtain a MCMC sample from this posterior, we first augment the state space by the parameters $(\boldsymbol{\rho}, \boldsymbol{\phi})$. Then, factorizing the augmented posterior as in (13), we can hierarchically simulate from $\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\phi}|data)$ first, and secondly apply the DBSCAN function to $(\boldsymbol{\rho}, \boldsymbol{\phi})$. There is plenty of methods to sample from the posterior law $\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\phi}|data)$ when the mixing measure is a species sampling model. In particular, we refer to the MCMC algorithm in Argiento et al. (2009), that extends the Gibbs sampler algorithms under the DPM model in Neal (2000).

In the Bayesian nonparametric model-based context, the choice of a suitable point estimate $\hat{\boldsymbol{\rho}}_\epsilon$ of the random partition $\boldsymbol{\rho}_\epsilon$ is a key point. By the way, in this context, by cluster analysis we actually mean any proper summary of the posterior distribution of $\boldsymbol{\rho}_\epsilon$. From a computational point of view, once we have obtained a MCMC sample from the posterior law $\mathcal{L}(\boldsymbol{\rho}_\epsilon|data)$, a Bayesian estimate of $\boldsymbol{\rho}_\epsilon$ is evaluated as a summary of the latter sample. Nevertheless, in general to find such a posterior estimate is a difficult task due to two issues. In fact, the support of the posterior distribution of $\boldsymbol{\rho}_\epsilon$ is a discrete space with large cardinality (the Bell number), so that evaluation of the posterior distribution in all the support points is computationally unfeasible. Furthermore, this space has a complex topology that does not allow to uniquely define a standard distance between two partitions. There exist many papers, in the Bayesian literature, dealing with these problems. Among the others, we refer to Quintana and Iglesias (2003), Medvedovic et al. (2004), Lau and Green (2007), Dahl (2009), Fritsch and Ickstadt (2009). Most of them follow this approach: a suitable loss function $L(\boldsymbol{\rho}_\epsilon, \hat{\boldsymbol{\rho}}_\epsilon)$ is fixed, giving the cost of estimating the "true" $\boldsymbol{\rho}_\epsilon$ by $\hat{\boldsymbol{\rho}}_\epsilon$. Then the proposed estimate is given by any $\hat{\boldsymbol{\rho}}_\epsilon$ which minimizes the posterior expectation of the loss function, i.e.

$$\hat{\boldsymbol{\rho}}_\epsilon \in \arg\min_y \mathbb{E}[L(\boldsymbol{\rho}_\epsilon, y)|data].$$

Following Binder (1978), we consider the loss function assigning cost b when two elements are wrongly clustered together and cost a when two elements are erroneously assigned to different clusters,

$$L(\boldsymbol{\rho}_\epsilon, \hat{\boldsymbol{\rho}}_\epsilon) = \sum_{i<j\leq n} \Big(a\, \mathbf{1}\{X_i \overset{\boldsymbol{\rho}_\epsilon}{\leftrightarrow} X_j, X_i \overset{\hat{\boldsymbol{\rho}}_\epsilon}{\nleftrightarrow} X_j\} + b\, \mathbf{1}\{X_i \overset{\boldsymbol{\rho}_\epsilon}{\nleftrightarrow} X_j, X_i \overset{\hat{\boldsymbol{\rho}}_\epsilon}{\leftrightarrow} X_j\Big) \tag{14}$$

where $\overset{\boldsymbol{\rho}_\epsilon}{\leftrightarrow}$ and $\overset{\hat{\boldsymbol{\rho}}_\epsilon}{\leftrightarrow}$ stand for the equivalence relations induced by the partitions $\boldsymbol{\rho}_\epsilon$ and $\hat{\boldsymbol{\rho}}_\epsilon$, respectively. It is not difficult to see (Lau and Green, 2007) that, if $\{s_{ij}\}$ is the matrix of the *posterior incidence probabilities* $\mathbb{P}(X_i \overset{\boldsymbol{\rho}_\epsilon}{\leftrightarrow} X_j|data)$ and $K = b/(a+b) \in [0,1]$, then the posterior mean of (14) can be written as

$$l(\hat{\boldsymbol{\rho}}_\epsilon) = a\sum_{i<j} s_{ij} - (a+b)\sum_{i<j} \mathbb{I}_{\{X_i \overset{\hat{\boldsymbol{\rho}}_\epsilon}{\leftrightarrow} X_j\}}(s_{ij} - K) = a\sum_{i<j} s_{ij} - (a+b)g(\hat{\boldsymbol{\rho}}_\epsilon) \tag{15}$$

Of course, minimizing $l(\hat{\boldsymbol{\rho}}_\epsilon)$ corresponds to maximizing $g(\hat{\boldsymbol{\rho}}_\epsilon)$, with respect to $\hat{\boldsymbol{\rho}}_\epsilon$. However, $\{s_{ij}\}$ is unknown. Lau and Green (2007) proposed a sophisticated optimization method considering a binary integer programming problem. Rather, as suggested by the two authors themselves, we used a simpler method: we ran the MCMC chain once in order to estimate the posterior probabilities $\{s_{ij}\}$, then we plugged this estimate in (15) and ran the MCMC algorithm a second time, obtaining a posterior sample configurations. Finally, as $\hat{\boldsymbol{\rho}}_\epsilon$, we chose the configuration, among the latter sampled ones, that maximize $g(\hat{\boldsymbol{\rho}}_\epsilon)$. Of course, the result is affected by the choice of the parameter K, which can be seen as the proportion of the cost to pay by putting together two elements, when they should be separated. In this work, $K = 0.5$ is fixed, so that the two costs are equally shared.

5 Clustering validation techniques

It is clear that one of the main issues in our approach is the choice of hyperparameter ϵ. As the application sections will show, this hyperparameter strongly affects the posterior cluster estimate. On the other hand, when ϵ is random, it is not straightforward to design an algorithm for posterior computation. Here we propose to fix ϵ in order to optimize some suitable posterior functionals. Our approach will be the following:

(a) Fix a grid of values $\epsilon_1, \dots, \epsilon_r$

(b) Evaluate the posterior expectation $\mathbb{E}(\mathcal{H}(\boldsymbol{\rho}_{\epsilon_j})|data)$ for a suitable function $\mathcal{H}$ for $j = 1, \dots, r$

(c) choose the optimal ϵ_j among $\epsilon_1, \dots, \epsilon_r$.

In order to introduce suitable functions $\mathcal{H}$ we will refer to cluster validation techniques literature. By such procedures we mean techniques comparing the quality assessment of the clustering estimates; see Halkidi et al. (2001) for a nice survey. We will use two popular such tools: the silhouette and the adjusted Rand indexes. To keep the description self-contained, we briefly review their definitions here.

The Silhouette Coefficient

The silhouette coefficient or index (Rousseeuw, 1987) evaluates the quality of a partition using only quantities and features inherent to the dataset. Given a distance (or a similarity) among the data and a partition $\boldsymbol{\rho} = \{C_1, \ldots, C_k\}$ of them, the following steps explain how to compute the silhouette coefficient for an individual point. First, for the $i-$th data, calculate the sample mean of the distance between the data and all the other in its cluster. Call this value a_i. Secondly, compute the sample mean of the distances between the $i-$th data and all the points in a cluster not containing it. Find the minimum such value with respect to all clusters; call this value b_i. Finally, for the $i-$th object, the silhouette coefficient is defined as $s_i = (b_i - a_i)/\max(a_i, b_i)$.

The value of the silhouette coefficient can vary between -1 and 1. It quantifies how good an observation fits its cluster: specifically, the largest is the value, the "better" the observation has been assigned to the "right" cluster. Indeed, if $a_i = 0$, the silhouette coefficient of the i-th observation is equal to 1. Moreover, a negative value is undesirable because this corresponds to a case in which a_i, the average distance to points in the cluster, is greater than b_i, the minimum average distance to points in another cluster. An overall measure of the quality of a partition can be obtained by computing the average silhouette coefficient of all points. We mention that, since the silhouette coefficient is not defined when there is a unique cluster, in this case we set it equal to 0.

Adjusted Rand Index

Differently from the silhouette coefficient, the adjusted Rand index (Hubert and Arabie, 1985) quantifies the difference among two given partitions. It is widely used in cluster validation analysis, when a "true" reference partition is available. Given a set of n elements and two partitions to compare, $\boldsymbol{\rho}_1 = \{C_1, \ldots, C_k\}$, $\boldsymbol{\rho}_2 = \{B_1, \ldots, B_s\}$, consider the following quantities: a, the number of pairs of elements that are in the same set in $\boldsymbol{\rho}_1$ and in the same set in $\boldsymbol{\rho}_2$; b, the number of pairs of elements that are in two different sets in $\boldsymbol{\rho}_1$ and in two different sets in $\boldsymbol{\rho}_2$; c, the number of pairs of elements that are in the same set in $\boldsymbol{\rho}_1$ but in different sets in $\boldsymbol{\rho}_2$; d the number of pairs of elements that are in different sets in $\boldsymbol{\rho}_1$ but in the same set in $\boldsymbol{\rho}_2$. The Rand index (Rand, 1971) is defined as:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}.$$

Keeping in mind that $a + b$ is the number of agreements between $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$, while $c + d$ is the number of disagreements, intuitively, R is the proportion of agreements between the two partition $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$. With the intent of making the values of this index more interpretable,

$\boldsymbol{\rho}_1 \backslash \boldsymbol{\rho}_2$	B_1	B_2	$\dots$	B_s	
C_1	n_{11}	n_{12}	$\dots$	n_{1s}	a_1
C_2	n_{21}	n_{22}	$\dots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
C_k	n_{k1}	n_{k2}	$\dots$	n_{ks}	a_k
	b_1	b_2	$\dots$	b_s	

Table 1: Contingency table summarizing the overlap between $\boldsymbol{\rho}_1 = \{C_1, \dots, C_k\}$ and $\boldsymbol{\rho}_2 = \{B_1, \dots, B_s\}$.

Hubert and Arabie (1985) introduced a correction based on a standardization (*correction for chance*). Suppose that the two partition $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$ to be compared are chosen according a generalized hypergeometric distribution, i.e. $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$ are picked at random, with fixed numbers of classes and objects in both. The authors define the adjusted Rand index as

$$AR = \frac{R - \mathbb{E}(R)}{\max(R) - \mathbb{E}(R)};$$

moreover they show that, under the generalized hypergeometric assumption,

$$AR = \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}},$$

where n_{ij}, a_i, b_j are those in Table 1. More precisely, the overlap between $\boldsymbol{\rho}_1 = \{C_1, \dots, C_k\}$ and $\boldsymbol{\rho}_2 = \{B_1, \dots, B_s\}$ can be summarized by a contingency table $[n_{ij}]$, where each entry n_{ij} denotes the number of objects in common between C_i and B_j, i.e. $n_{ij} = \#\{C_i \cap B_j\}$; the values a_i and b_j denote the marginal frequencies, respectively.

A model-based predictive index

The indexes introduced so far are not completely satisfactory. On one hand, the adjusted Rand index needs a "true" partition as a reference. Generally, in real applications this "true" partition does not exist. On the other hand, the silhouette index needs a distance between data to be computed; however, our approach to clustering is based on the notion of distance between densities depending on latent variables. For this reason, we will compute these two indexes only in the simulated data example in Section 6, where we know the "true" partition. For real applications as in Section 7, we introduce an index built from predictive distributions under our model.

Let X_{new} be a new observation from (1)-(2). For $i = 1, \dots, n$, let

$$Y^{\epsilon}_{new}(X_i) = \begin{cases} 1 & \text{if } X_{new} \text{ is in the same cluster of } X_i \\ 0 & \text{otherwise.} \end{cases}$$

In other words, $Y^{\epsilon}_{new}(X_i)$ tells us if the new observation X_{new} belongs to the same cluster where X_i is. Therefore, for each i, we consider $\mathcal{L}(X_{new}|Y^{\epsilon}_{new}(X_i) = 1, data)$, that is the predictive law of a new observation conditionally to the event that this observation share the same cluster with X_i.

In the same spirit as in Gelfand et al. (1992), for a fixed ϵ, we compute conditional predictive residuals defined as

$$r_i^{(\epsilon)} := r_i = \frac{X_i - \mathbb{E}(X_{new}|Y^{\epsilon}_{new}(X_i) = 1, data)}{(\mathrm{Var}(X_{new}|Y^{\epsilon}_{new}(X_i) = 1, data))^{1/2}}. \qquad i = 1, ..., n. \tag{16}$$

When the data are multivariate, the square root of the matrix in the denominator in (16) represents its Cholesky decomposition. For each data component $j = 1, \ldots, p$, we compute

$$\mathrm{Ind}_j^{(\epsilon)} := \frac{1}{n}\sum_{i=1}^{n} r_{i,j}^2, \tag{17}$$

which represents a predictive goodness-of-fit index of our DBSCAN-mixture model on the $j-$th data component as a function of ϵ.

Moreover, we compute the following predictive probabilities, for any fixed $\epsilon > 0$:

$$\mathbb{P}(Y^{\epsilon}_{new}(X_i) = 1|X_{new} = X_i, data), \quad i = 1, \ldots, n. \tag{18}$$

In words, for each i, (18) is the probability that a new observation is assigned to the same cluster as X_i, conditionally to the event that X_{new} and X_i assume exactly the same value. However, for a fixed i, the value assumed by such an index cannot be interpreted "per se", but it must be compared to all the other values ($j \neq i$). High values denote that X_i is "nested" in its cluster, while small values suggest that X_i is a "frontier" point in the cluster it has been assigned to. Hence, those probabilities have an interpretation as misclassification indexes.

The Appendix shows how to compute (16) and (18) through a posterior sample of $(\boldsymbol{\rho}, \boldsymbol{\phi})$.

6 Simulated data

In this section we illustrate our model with application to a simulated dataset of size $n = 1000$. In particular, we simulated i.i.d. observations from a mixture of bivariate densities. Data are shown in Figure 1; there are two main groups of observations, from the two components of the mixture: the first one has a sharp round shape and it is located around the point $(0, 0)$, while the second group lays on a semicircular region on the right of the first group. This peculiar disposition of the observations on a non-convex support is a popular choice when dealing with clustering algorithms, in order evaluate how well they perform even in unusual situations.

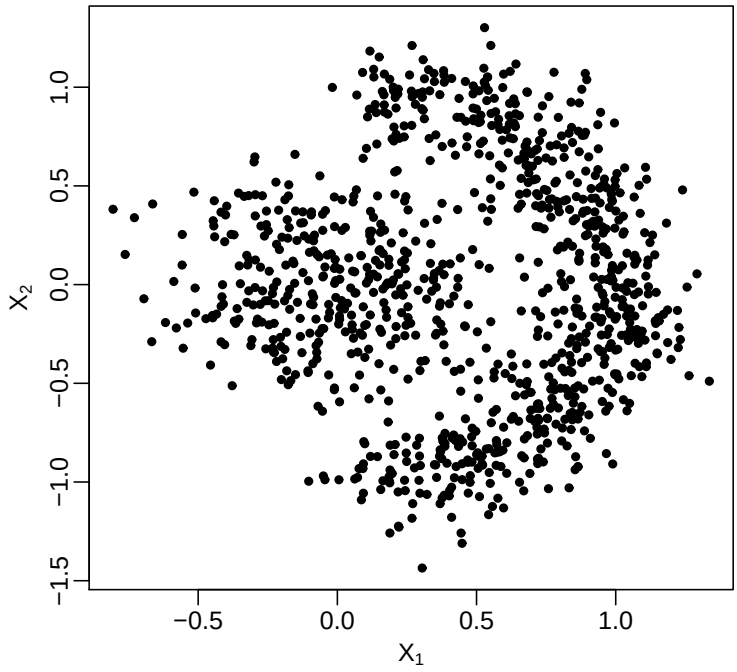

Figure 1: Simulated bivariate dataset.

As far as the b-DBSCAN mixture model (11) is concerned, we assume the Gaussian kernel for $f(\cdot;\theta)$, while the mixing measure is the Dirichlet process. In particular, we complete the prior specification by assuming a prior $\alpha \sim \text{gamma}(\gamma_1, \gamma_2)$, while $P_0(d\theta) = N(d\mu|m_0, \Sigma/\kappa_0) \times \text{Inv-Wishart}(d\Sigma|\nu_1, \Psi_1)$, where $\theta = (\mu, \Sigma)$. Here $\text{gamma}(\gamma_1, \gamma_2)$ denotes the univariate gamma distribution with mean γ_1/γ_2 and $\text{Inv-Wishart}(\nu_1, \Psi_1)$ represents the Inverse-Wishart distribution having ν_1 degrees of freedom and precision matrix Ψ_1 (and $\mathbb{E}(\Sigma) = \Psi_1/(\nu_1 - p - 2)$). First of all, we fixed $\epsilon = 0$, that is when model (11) reduces to a DPM model. As far as the hyperparameters are concerned, we did a robustness analysis, choosing different values for γ_1, γ_2, m_0, κ_0, ν_1 and Ψ_1. We will not report these analyses here, but we would like to point out that the conclusions on the cluster estimates are always the same: the estimated number of clusters is larger than the true one, that is 2. This is an expected result, since many Gaussian densities are needed to fit the non-convex region on the right of the plot in Figure 1. On the other hand, when ϵ is larger than 0, to make the b-DBSCAN model more flexible, it is better to fix hyperparameters so that the conditional variance of $f(\cdot;\theta)$ is small, and the prior expected number of mixture components is large. In particular, we fixed a and b such that $E(\alpha) = 11$, $\text{Var}(\alpha) = 4$, $m_0 = 0$, $k_0 = 0.001$, $\nu_1 = 10$ and $\Psi_1 = \text{diag}(0.1)$. Figure 2 displays the incidence matrices of the estimated clusters for

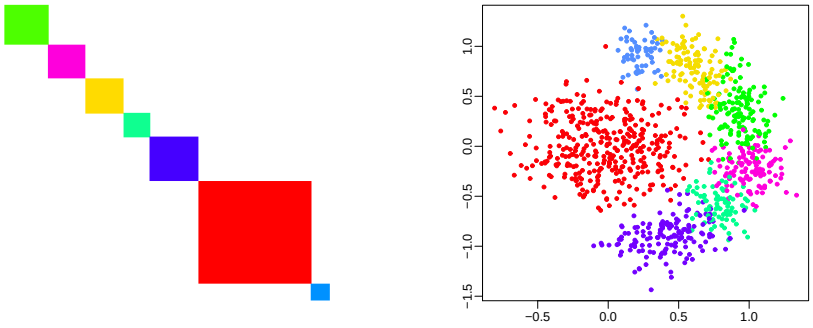

Figure 2: Cluster estimation when $\epsilon = 0$ (i.e. DPM model): incidence matrix (left) and classification of the data (right).

$\epsilon = 0$ on the left and the corresponding data clustering representation on the right. By an incidence matrix M we mean a $n \times n$ matrix whose entries m_{ij} records whether two observations are clustered together ($m_{ij} = 1$) or not ($m_{ij} = 0$). Moreover, to produce summary plots, we order rows and columns of the incidence matrices according their membership to clusters, assigning them different colours; of course, only the elements with positive entries are shown. The incidence matrix in all the figures here is always followed by a plot of the dataset, where each observation is coloured according to the estimated group it belongs to.

As discussed in Section 3, to define the b-DBSCAN model, we need to to fix a distance $d(\cdot, \cdot)$ between distribution. Here we use Hellinger, L^2 distances and Kullback-Leibler I-divergence (it can be symmetrized to become a pseudo-distance). Figures 3, 4 and 5 show the estimates for different values of ϵ under these distances. As we expected, the estimated number of clusters reduces as ϵ increases in Figures 3, 4, 5: in fact, the model groups the DPM clusters into new bigger clusters. The choice of the distance can greatly affect the posterior cluster estimate: for the Hellinger distance and Kullback-Leibler I-divergence, as ϵ increases groups with similar mean parameters are merged, and we find good posterior estimates (third column of Figures 3 and 4). In contrast, under L^2 distance, as ϵ increases groups with similar covariance matrix are merged, and in this case the clusters follow a different grouping path, leading to unsatisfactory estimated partitions.

The choice of ϵ and misclassification

We mentioned many times that the choice of hyperparameter ϵ is the most difficult task in our model. Let us see how we fixed it in this application, for instance when d is the symmetrized Kullback-Leibler I-divergence. Following the scheme outlined at the beginning

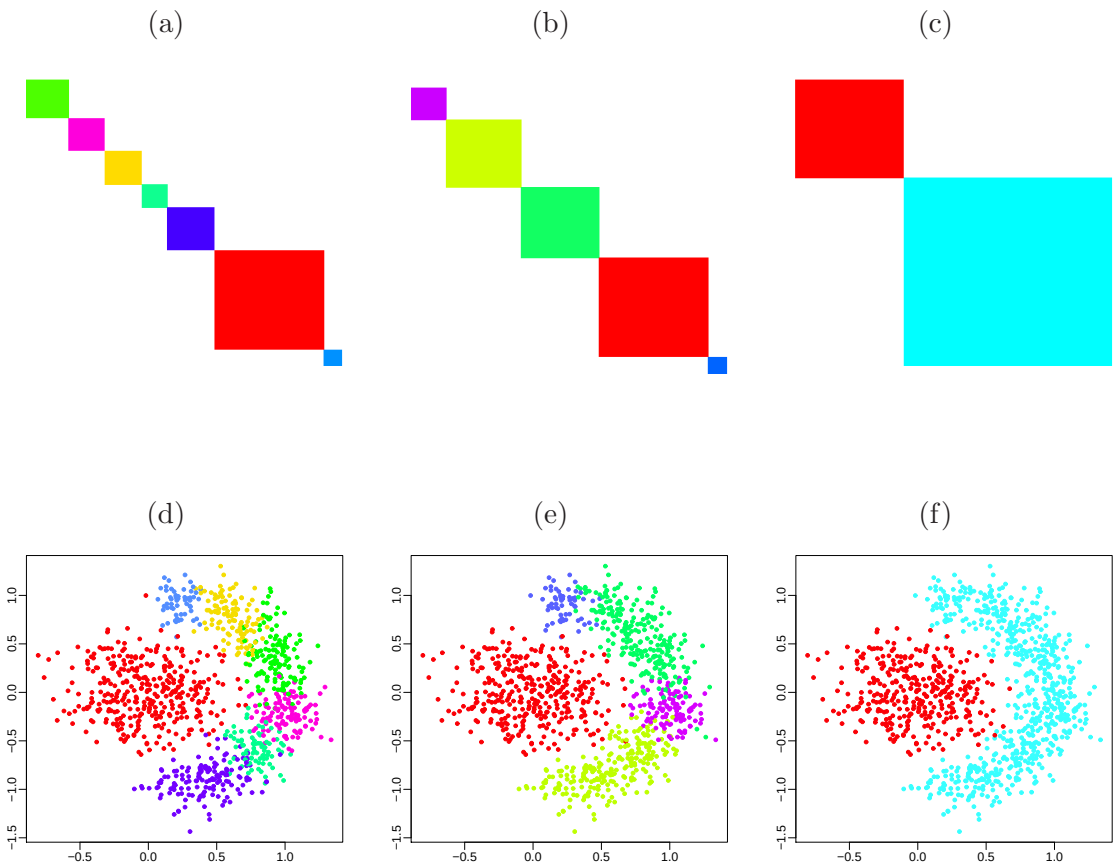

Figure 3: Posterior estimate of the random partition, i.e. incidence matrix and data cluster membership, when d is the Hellinger distance, for $\epsilon = 0$ (panels (a) and (d)), $\epsilon = 0.8$ ((b) and (e)), $\epsilon = 0.9$ ((c) and (f)).

of Section 5, we fixed a grid of values of ϵ. On the log-scale we chose these values: $\log(1+\epsilon) \in \{0.5, 1.5, 2, 2.5, 2.75, 3, 3.5, 4\}$. For each $j = 1, \ldots, 8$ we computed $\mathbb{E}(\mathcal{H}(\boldsymbol{\rho}_{\epsilon_j})|data)$ through the MCMC method, where $\mathcal{H}$ is the silhouette or the adjusted Rand index (here we know the true data partition). Figure 6(a) shows the two posterior functionals, as ϵ varies. Both lines lead to the same conclusion: $\log(1 + \epsilon) = 3$ is the optimal choice. Figure 4 (right column) shows that, under this choice, our estimate is very close to the true partition.

Table 2 reports a summary of the misclassification error when the distance is the Kullback-Leibler I-divergence, under the optimal (i.e. $\log(1 + \epsilon) = 3$) estimated partition. To simplify the discussion, let us call cluster A the one with round shaped support on the left of Figure 1 and cluster B the other one. We found that 337 points in cluster A, and 644 points in cluster B, were correctly classified; the misclassification rate is 1.9%. Moreover we computed the misclassification probability index in (18) for each data points. Since all the individual values of this index should be compared to the others in order to use it meaningfully, we first

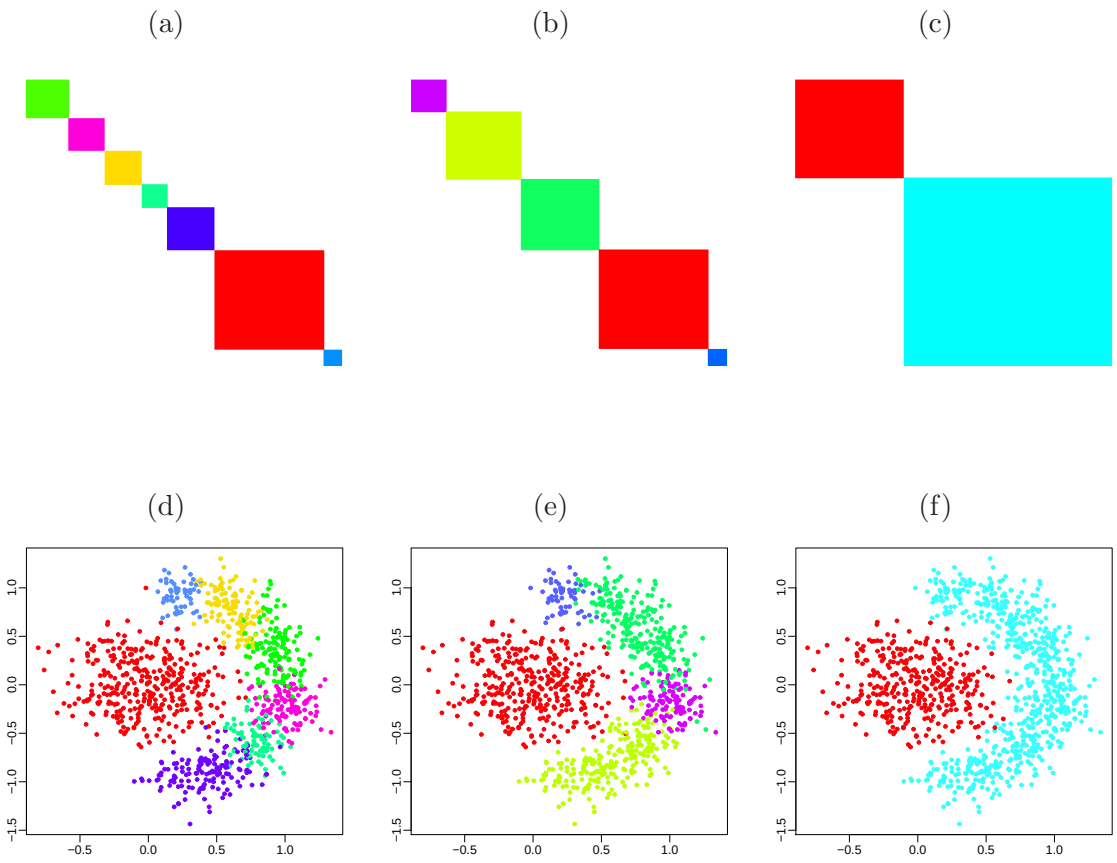

Figure 4: Posterior estimate of the random partition, i.e. incidence matrix and data cluster membership, when d is the symmetrized Kullback-Leibler I-divergence, for $\epsilon = 0$ (panels (a) and (d)), $\log(1+\epsilon) = 2.5$ ((b) and (e)), $\log(1+\epsilon) = 3$ ((c) and (f)).

computed $q_{.25}$ and $q_{.75}$, the first and the third sample quantile of the values of the index. We classified as boundary points all the data such that the corresponding probability index is smaller than $q_{.25} - 1.5(q_{.75} - q_{.25})$. A summary of the results is depicted in Figure 6(b), where the boundary points are represented by (red) triangles, while misclassified data are represented by (blue) crosses. Observe as misclassified data lie in the middle of the two main groups, where there is uncertainty between the two clusters membership. Moreover, there is an area of cluster membership uncertainty on the left side of cluster A. For these points the uncertainty is between the membership to cluster A or to a new cluster not included in the estimated ones.

DBSCAN algorithm

As described in the Introduction, the DBSCAN algorithm is a heuristic clustering method that unifies elements close to each other, and is able to locate dense group of observations.

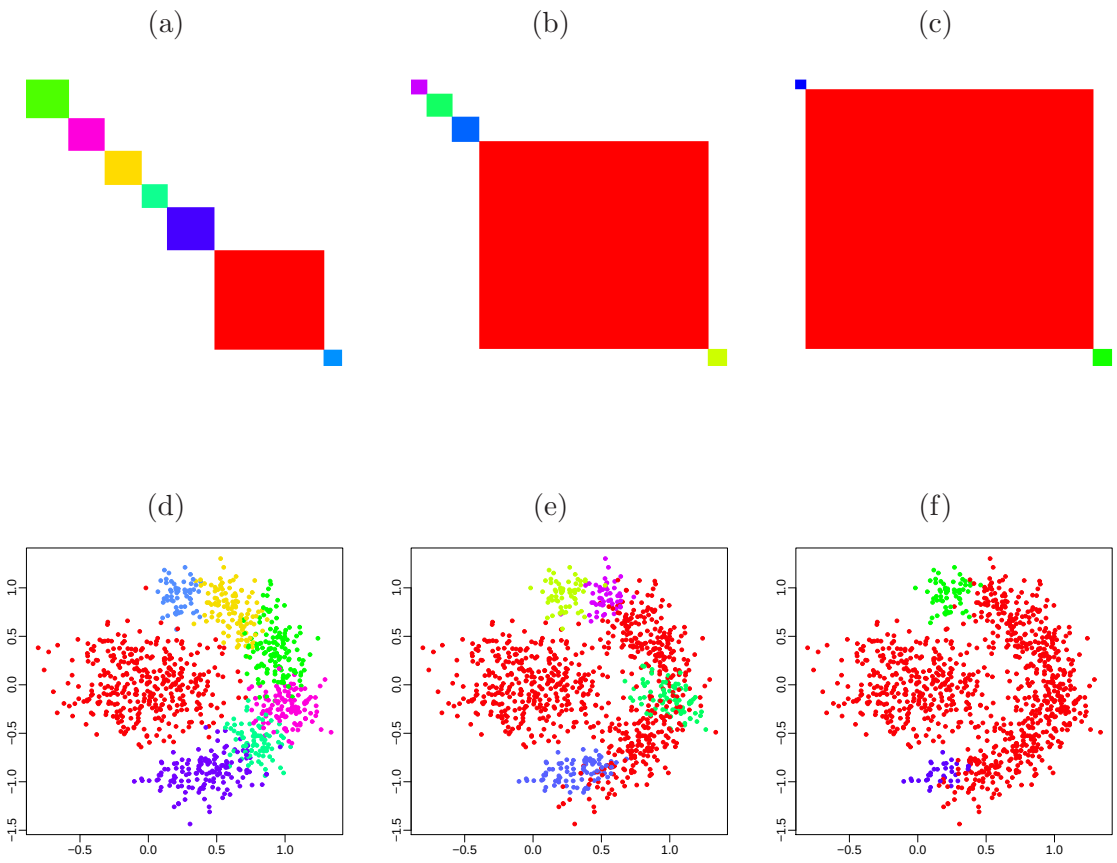

Figure 5: Posterior estimate of the random partition, i.e. incidence matrix and data cluster membership, when d is the L^2 distance, for $\epsilon = 0$ (panels (a) and (d)), $\epsilon = 1.925$ ((b) and (e)), $\epsilon = 2.2$ ((c) and (f)).

Cluster membership	Estimated A	Estimated B
True A	337	13
True B	6	644

Table 2: Summary of the true and estimated clustering, i.e. 337 points belonging to cluster A were correctly classified, and 664 points belonging to cluster B were correctly classified.

For the aim of comparison, here we would like to directly cluster the simulated data using this procedure, not resorting to the corresponding latent variables in the b-DBSCAN model, fixing d as the Euclidean distance among points in $\mathbb{R}^2$. Moreover, we fixed N equal to 1, but also larger than 1. Recall that, when $N > 1$, the partitions are not uniquely determined, because the relation defined among the data is not an equivalence. Furthermore, when $N > 1$, noise elements are usually identified by the algorithm. In Figure 7 we report clustering results for

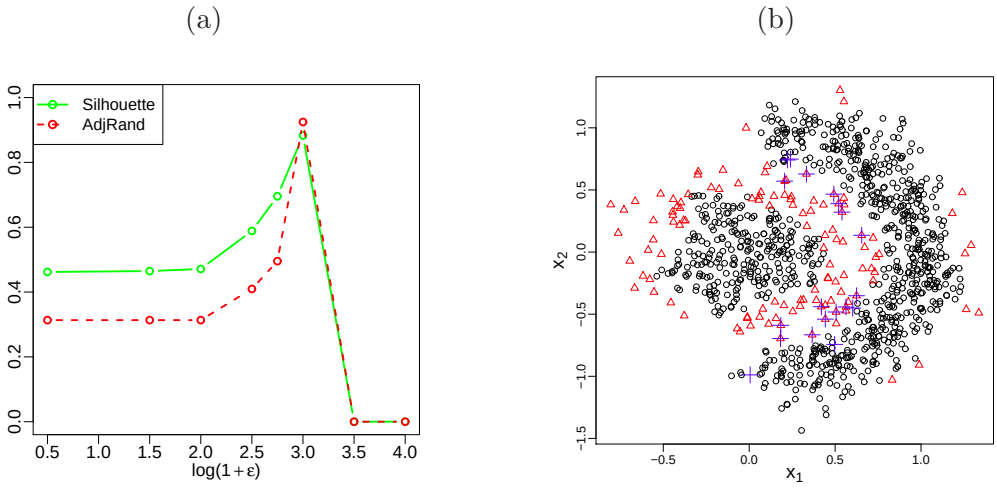

Figure 6: Panel (a) Posterior expectation of the adjusted Rand and silhouette functionals under various choice of $\log(1+\epsilon)$ (panel (a)) and misclassification graph (panel (b)).

two different values of (N, ϵ), choosing these two pairs among those better reflecting the true partition.

When $N = 1$ (Figure 7, left column), noise elements are not allowed, and every singleton could represent a cluster. This is the reason why so many different clusters are identified by the method. Of course, this partition does not seem to be satisfactory, if compared to b-DBSCAN estimates. In contrast, when $N = 6$, less clusters are found, but many points are classified as "noise" by the algorithm. See the red points in Figure 7(d): they correspond to the red square on the bottom of the incidence matrix in panel (b), but do not form a cluster. The main reason why this happens is that the heuristic DBSCAN algorithm does not need any model to be defined, and hence points generated from the tails of the true distribution are not included into the clusters. Finally, as an example of the non-uniqueness of the partition found by the heuristic method when $N > 1$, consider the triangle blue points (just above the red central group), which are classified as a unique cluster. This group contains only three points, despite that $N = 6$ is the minimum number of points to define a cluster. The ambiguity arises since, in this case, N is larger than 1, so that the clustering produced by the heuristic DBSCAN is not uniquely defined.

7 Yeast cell cycle data

We fitted our model to a dataset, very popular in the literature on clustering of gene expression profiles, usually called Yeast cell cycle data (see Cho et al., 1998, for instance). A gene expression data set from a microarray experiment can be represented by a real-valued matrix

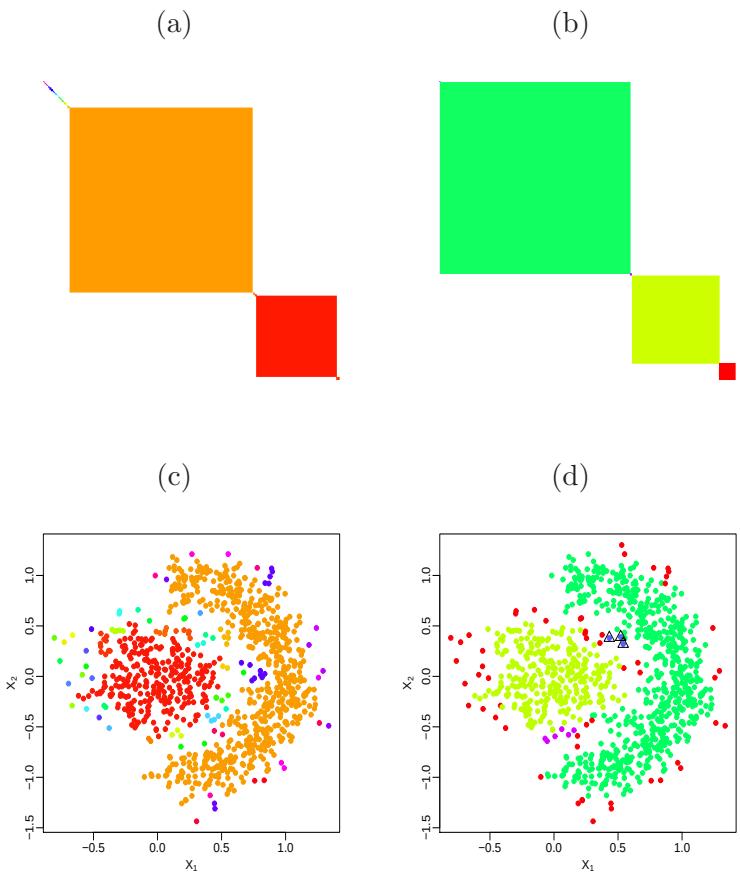

Figure 7: Heuristic DBSCAN clustering results of the simulated dataset when $N = 1$, $\epsilon = 0.075$ (panel (a) and (b)) and $N = 6$, $\epsilon = 0.1$ (panel (b) and (d)).

$[X_{ij},\ 1 \leq i \leq n, 1 \leq j \leq p]$, where the rows $(X_1, \ldots, X_n)$ contain the expression patterns of genes and will be our data points. Each cell X_{ij} is the measured expression level of gene i in sample (or at time) j. The Yeast cell cycle data contain $n = 389$ gene expression profiles, observed at 17 different time values, one every 10 minutes from time zero. We chose a subset of the original dataset, representing the second cell cycle. The final dataset ($n = 389$, $p = 9$) has been obtained by a filter, i.e. standardizing each row of the gene expression matrix to have zero mean and unit variance. By visual inspection Cho et al. (1998) grouped the data according the peak times of expression levels; see Figure 8. They detected five peaking points, corresponding to five phases of the cell cycle: the early G1 phase at time $j = 10$, the late G1 phase at time $j = 11$, the late S phase at time $j = 12$, the G2 phase at time $j = 14$ and the M phase at time $j = 16$. This clusterization can be considered as a reference partition. However, we would like to stress that, since this clusterization was obtained by visual inspection, it could be dramatically affected by subjective belief. For this reason, we will not consider this

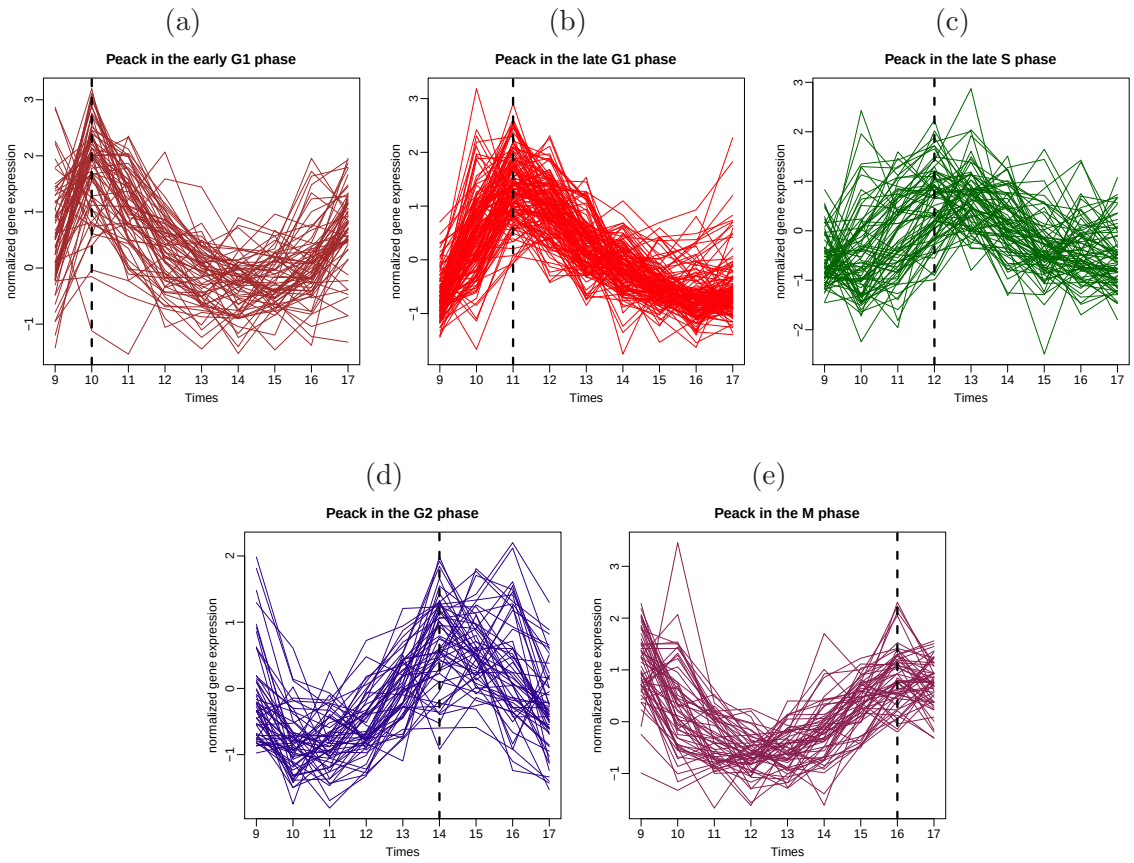

Figure 8: Reference partition by Cho et al. (1998) for the Yeast cell cycle data.

partition as the "true" one, but rather as as benchmark to compare with our results.

As in the previous example, we assume the Gaussian kernel as $f(\cdot;\theta)$ and the Dirichlet process as mixing measure. The latent variable here is $\theta = (\mu, \sigma^2\mathbb{I}_p)$ representing mean and covariance matrix of the Gaussian distribution. Moreover, conditionally on the total mass parameter α, $P \sim \text{Dirichlet}(\alpha, P_0)$, with $\alpha \sim \text{gamma}(\gamma_1, \gamma_2)$, and $P_0(d\mu, d\sigma^2) = N(d\mu|m_0, \sigma^2/\kappa_0\mathbb{I}_p) \times \text{inv-gamma}(d\sigma|a, b)$. Observe that, following the work of Qin (2006), the Gaussian kernel densities were chosen to have diagonal covariance matrices. This assumption greatly simplifies computation, since only diagonal matrices must be inverted in the MCMC algorithm. On the other hand, under this assumption, data are modelled from a mixture of Gaussian kernels with spherical contour lines. This assumption is very strong when $\epsilon = 0$: it implies that all clusters have spherical shapes a priori. However, this is not the case under the b-DBSCAN model for $\epsilon > 0$, where the clusters are modelled as finite unions of round shaped groups, and therefore they can recover many different shapes.

As far as the choice of the hyperparameters is concerned, in order to make the model more flexible, we fixed them so that the prior number of mixture components is large. In particular,

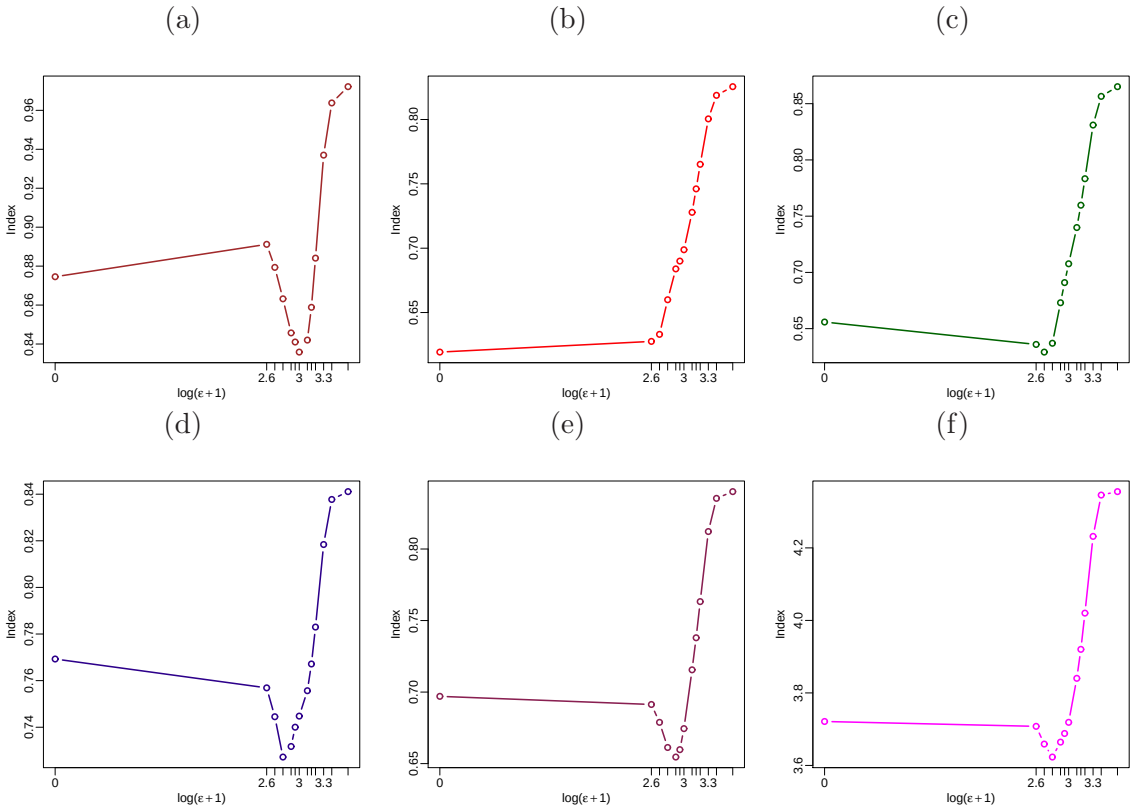

Figure 9: Plots of $\text{Ind}_j^{(\epsilon)}$, $j \in \{10, 11, 12, 14, 16\}$, and their cumulative value for the Yeast cell cycle data.

we fixed (m_0, κ_0, a, b) so that the prior variance for μ is large $(10\mathbb{I}_p)$, but the prior mean and variance of σ^2 are small (both equal to 0.1). Furthermore, we set $(\gamma_1, \gamma_2) = (2, 0.01)$, in order to obtain a vague prior for the total mass parameter α.

To complete the b-DBSCAN specification we need to fix the distance and the threshold ϵ. In our experiments we considered the Hellinger distance, as well as the Kullback-Leibler I-divergence, obtaining very similar results. Here we report only the analysis under the Kullback-Leibler I-divergence. As far as the choice of ϵ is concerned, we applied the strategy described in Section 5, using the index (17). We fixed the following grid of values: $\log(\epsilon+1) \in \{0, 2.6, 2.7, 2.8, 2.9, 2.95, 3, 3.1, 3.15, 3.2, 3.3, 3.4, 3.6\}$. In particular we computed $\text{Ind}_j^{(\epsilon)}$ for each $j \in \{10, 11, 12, 14, 16\}$, which are the times at which the data have peaks according to Cho et al. (1998). As we can see from Figure 9, except for the late G1 phase (panel (b)), all the index trajectories have a minimum around $\epsilon = 2.8$; furthermore, the trajectories of the sum of the indexes (see panel (f)) has a minimum exactly at $\epsilon = 2.8$. Figure 10 shows our cluster estimate for such a value of ϵ. Observe that we found 8 clusters, a number larger than the five clusters in the reference partition in Figure 8. However, the reference partition is based only

on peak times of the five cell cycle phases, so that it could not be able to capture the patterns of the gene expression across time. On the other hand, our clusterization takes into account

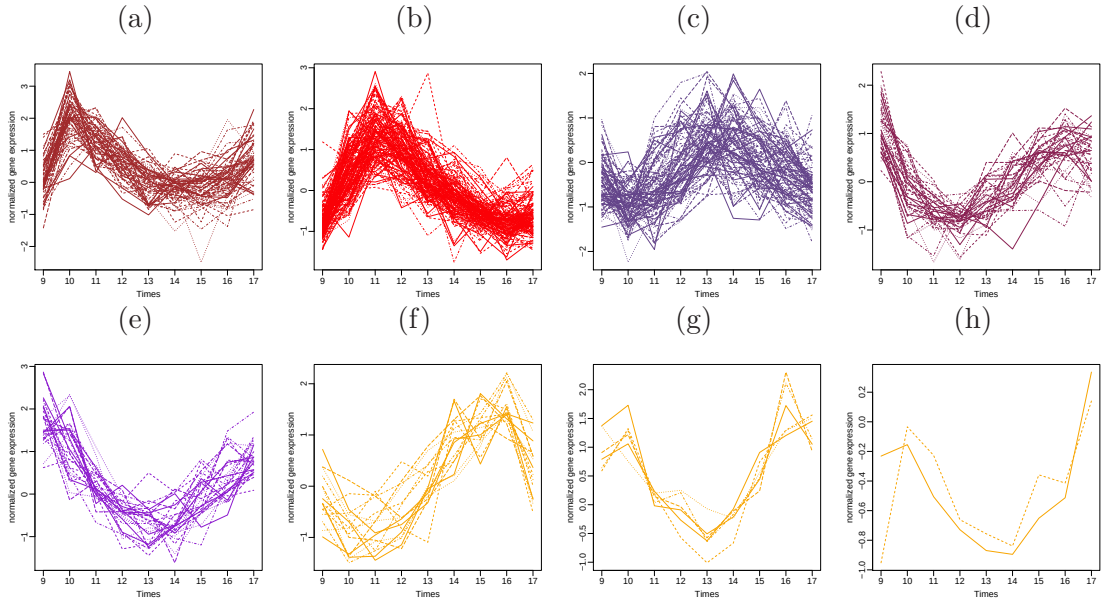

Figure 10: Posterior cluster estimate for the Yeast cell cycle data.

not only the different peaks, but also the entire trajectories of gene expression. For example, Figure 10 shows that the cluster in panel (a) groups trajectories with a peak in the early G1 phase, while the cluster in panel (b) groups those with a peak in the late G1 phase. The correspondence between the reference and our partitions is not so unequivocal for the other groups: for example the cluster in Figure 10(c) puts together trajectories with peaks in the late S phase or G2 phase. Moreover, according to our estimated partition, the trajectories in each groups seem more homogeneous. In particular, our algorithm is able to split the late S phase group in Figure 8 into more homogeneous clusters, in terms of trajectories. As a final remark, note that a positive feature of our procedure is the ability to classify non-standard data: see the "outlier" trajectories grouped into two clusters 7 and 8 (Figure 10, panels (g) and (h)).

We have also checked robustness of these results to choices of hyperparameters; for brevity, this analysis is not reported here.

8 Comments

We have presented a Bayesian nonparametric framework for model-based clustering. Data have been initially modeled through a species sampling mixture model. The core of our work lies in defining the data partition parameter in a new way: two observations are in

the same cluster if the distance between densities corresponding to their latent parameters is smaller than a threshold ϵ. This definition is made mathematically coherent introducing the *reachability* property in Definition 1 and 2. We call the proposed model b-DBSCAN mixture. This model can be interpreted as a mixture whose components are themselves mixtures of parametric densities (for instance, Gaussian kernels). Crucial ingredients are the (pseudo-)distance d between densities, and the hyperparameter ϵ.

We discussed implementation and applications of the b-DBSCAN mixture model to two datasets. The first one is simulated from a mixture of two components, one of them being with curved support. The second dataset is well-known in the literature of clustering gene expression data; each observation represents a trajectory over time of gene expression. From our analysis we conclude that the b-DBSCAN mixture model is affected by the choice of the distance between densities. In fact, when ϵ is fixed, Kullback-Leibler I-divergence (or Hellinger distance) and L^2 distance give very different estimates. In particular, we have observed that clusters with *centers* close to each other are grouped (more and more as ϵ increases) when the distance is Kullback-Leibler I-divergence or Hellinger. On the other hand, robust features of the estimates hold with respect to the choice of the hyperparameters of the baseline P_0, of the total mass α, and the parameter K (the proportion of misclassification costs). See the extensive robustness analysis in Cremaschi (2012). As far as the elicitation of ϵ is concerned, we suggested a strategy to fix it, as the optimal value of the posterior expectation of a function of the random partition. For the Yeast cell cycle data, we computed the cluster estimates based on a predictive fit index. They results particularly satisfactory, since they have some features in common with the reference partition on one hand, while grouping data into more homogeneous clusters in terms of trajectories.

As a further remark, we point out that we made a comparison between our estimates and more standard heuristic algorithms, as hierarchical or k-means clustering, for the simulated dataset. Apart from the statistical advantages of model-based methods (estimating missing data, or taking into account the "randomness nature" of the data), we found that the heuristic approaches provide estimates far from the true partition (see Cremaschi, 2012). Here, we have reported only the clustering obtained using the standard heuristic DBSCAN procedure, which still provides unsatisfactory grouping.

In the two applications here, we have always assumed $\sigma = 0$ in the underline NGG process (i.e. the mixing measure), indeed obtaining a DPM model. The interested reader should refer to Cremaschi (2012) for an application with $\sigma > 0$.

Finally, extensions to the current approach include further work on the elicitation of ϵ and categorical formulations of this clustering model. These and other topics are the subject of current research.

Appendix

Computation of (16) *and* (18).

In this Appendix we fix the hyperparameters of the mixing distribution P, i.e. σ, α, P_0. When some of them are random, the reader can easily understand how the following calculations modify.

Let us start observing that, in order to compute the conditional mean and variance in (16), we must evaluate the posterior distribution of X_{new}, conditionally to the event $\{Y^{\epsilon}_{new}(X_i) = 1\}$:

$$\mathbb{P}(X_{new} \in dx | Y^{\epsilon}_{new}(X_i) = 1, data) = \frac{\mathbb{P}(X_{new} \in dx, Y^{\epsilon}_{new}(X_i) = 1 | data)}{\mathbb{P}(Y^{\epsilon}_{new}(X_i) = 1 | data)} \tag{19}$$

Analogously, to compute (18) we need to evaluate

$$\mathbb{P}(Y^{\epsilon}_{new}(X_i) = 1 | X_{new} = dx, data) = \frac{\mathbb{P}(X_{new} \in dx, Y^{\epsilon}_{new}(X_i) = 1 | data)}{\mathbb{P}(X_{new} \in dx | data)} \tag{20}$$

Fractions (19) and (20) share the same numerator. If $\boldsymbol{\theta} := (\theta_1, \ldots, \theta_n)$, and θ_{new} is the latent variable associated to X_{new}, then it holds:

$$\begin{aligned}
&\mathbb{P}(X_{new} \in dx, Y^{\epsilon}_{new}(X_i) = 1 | data) = \int_{\Theta \times \Theta^n} \mathbb{P}(X_{new} \in dx, Y^{\epsilon}_{new}(X_i) = 1, d\theta_{new}, d\boldsymbol{\theta} | data) \\
&= \int_{\Theta \times \Theta^n} \mathbb{P}(X_{new} \in dx, Y^{\epsilon}_{new}(X_i) = 1 | \theta_{new}, \boldsymbol{\theta}, data) \mathcal{L}(d\theta_{new}, d\boldsymbol{\theta} | data) \\
&= \int_{\Theta \times \Theta^n} \mathbb{P}(X_{new} \in dx | \theta_{new}, \boldsymbol{\theta}, data) \mathbb{P}(Y^{\epsilon}_{new}(X_i) = 1 | \theta_{new}, \boldsymbol{\theta}, data) \mathcal{L}(d\theta_{new} | \boldsymbol{\theta}, data) \mathcal{L}(\boldsymbol{\theta} | data).
\end{aligned}$$

Keep in mind that, conditionally to θ_{new}, the future observation X_{new} does not depend on either $\boldsymbol{\theta}$ or the data, and it has density $f(\cdot; \theta_{new})$. Moreover the θ_{new} is independent from the data conditionally on $\boldsymbol{\theta}$. Finally, observe that the variable $Y^{\epsilon}_{new}(X_i)$ is a deterministic function of θ_{new} and $\boldsymbol{\theta}$. Consequently, we have

$$\mathbb{P}(X_{new} \in dx, Y^{\epsilon}_{new}(X_i) = 1 | data) = \int_{\Theta \times \Theta^n} f(x; \boldsymbol{\theta}_{new}) dx \mathbf{1}_{\{Y^{\epsilon}_{new}(X_i) = 1\}} \mathcal{L}(d\boldsymbol{\theta}_{new} | \boldsymbol{\theta}) \mathcal{L}(d\boldsymbol{\theta} | data).$$

Suppose now that $\boldsymbol{\theta}$ is a sample from $\mathcal{L}(\boldsymbol{\theta} | data)$, such that $\boldsymbol{\rho} = \{C_1, \ldots, C_k\}$ and $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_k)$ with $\boldsymbol{\rho}_{\epsilon} = \{C^{(\epsilon)}_1, \ldots, C^{(\epsilon)}_m\}$. Moreover, suppose that $i \in C^{\epsilon}_h = C_{l^h_1} \cup \cdots \cup C_{l^h_{k^{\epsilon}_h}}$, with $\{l^h_1, \ldots, l^h_{k^{\epsilon}_h}\} \subseteq \{1, \ldots, k\}$. If P in (3) is a NGG process, i.e. $P \sim NGG(\sigma, \alpha, P_0)$, then the predictive distributions $\mathcal{L}(\theta_{new} | \boldsymbol{\theta})$ can be represented as

$$P(\theta_{new} \in B | \boldsymbol{\theta}) = w_0(n, k; \sigma, \alpha) P_0(B) + w_1(n, k; \sigma, \alpha) \sum_{j=1}^{k} (n_j - \sigma) \delta_{\phi_j}(B), \tag{21}$$

where n_j is the cardinality of the j-th cluster of $\boldsymbol{\rho}$ and where w_0 and w_1 are predictive weights associated to the NGG process prior (see Lijoi et al., 2007, for an explicit expression). Now if θ_{new} is a sample from (21), it is clear that we have the following four alternatives:

(a) θ_{new} is one among the values $\phi_{l_1^h}, \ldots, \phi_{l^h_{k_h^\epsilon}}$, associated with the elements in C_h^ϵ. Then $X_{new} \overset{\epsilon}{\leftrightarrow} X_i$, and hence, conditionally to $\boldsymbol{\theta}$ and θ_{new}, we have $Y^\epsilon_{new}(X_i) = 1$;

(b) θ_{new} coincides with one of the $\phi_1, \ldots, \phi_k$ different from $\phi_{l_1^h}, \ldots, \phi_{l^h_{k_h^\epsilon}}$. Then $X_{new} \overset{\epsilon}{\nleftrightarrow} X_i$, and hence, conditionally on $\boldsymbol{\theta}$ and θ_{new}, we have $Y^\epsilon_{new}(X_i) = 0$;

(c) θ_{new} is a new value chosen according to $P_0(\cdot)$ such that $d(f(\cdot, \theta_{new}), f(\cdot; \phi_{l_j^h})) < \epsilon$ for some $j \in \{1, \ldots, k_h^\epsilon\}$, so that $X_{new} \overset{\epsilon}{\leftrightarrow} X_i$; then, conditionally on $\boldsymbol{\theta}$ and θ_{new}, we have $Y^\epsilon_{new}(X_i) = 1$;

(d) θ_{new} is a new value chosen according to $P_0(\cdot)$ such that $d(f(\cdot, \theta_{new}), f(\cdot; \phi_{l_j^h})) \geq \epsilon$ for all $j \in \{1, \ldots, k_h^\epsilon\}$, so that $X_{new} \overset{\epsilon}{\nleftrightarrow} X_i$; then, conditionally on $\boldsymbol{\theta}$ and θ_{new}, we have $Y^\epsilon_{new}(X_i) = 0$.

From these arguments, analytically integrating out θ_{new} (where possible), and by a change-of-variable in the integral, we have:

$$\begin{aligned}\frac{1}{dx}\mathbb{P}(X_{new} \in dx, &Y^\epsilon_{new}(X_i) = 1|data) \\ &= \int \Big\{ w_1(n_{l_1^h} - \sigma) f(x; \phi_{l_1^h}) + \cdots + w_1(n_{l^h_{k_h^\epsilon}} - \sigma) f(x; \phi_{l^h_{k_h^\epsilon}}) \\ &\qquad + \int w_0 f(x; \theta_{new}) \mathbf{1}_{\{Y^\epsilon_{new}(X_i)=1\}} P_0(d\theta_{new}) \Big\} \mathcal{L}(d\boldsymbol{\rho}, d\boldsymbol{\phi}|data).\end{aligned}$$

Now factor the mixing measure in the integral above as $\mathcal{L}(d\boldsymbol{\rho}, d\boldsymbol{\phi}|data) = \mathcal{L}(d\boldsymbol{\rho}|data) \times \mathcal{L}(d\boldsymbol{\phi}|\boldsymbol{\rho}, data)$; in addition, it holds that

$$\mathcal{L}(\boldsymbol{\phi}|\boldsymbol{\rho}, data) = \prod_{j=1}^{k} \mathcal{L}(\phi_j|\boldsymbol{X}_{C_j}) \propto \prod_{j=1}^{k} \left\{ \prod_{i \in C_j} f(x_i; \phi_j) P_0(d\phi_j) \right\}.$$

In practice, conditionally on partition $\boldsymbol{\rho}$ and data, $\phi_1 \ldots, \phi_k$ are independent, and the conditional law of ϕ_j depends only on data belonging to cluster C_j, $j = 1, \ldots, k$. The distribution $\mathcal{L}(\phi_j|\boldsymbol{X}_{C_j})$ represents the posterior of ϕ_j when $\pi(\cdot; p, P_0)$ in (3) is a degenerate prior on P_0 (parametric model), for $j = 1, \ldots, k$:

$$\begin{aligned}\{X_i,\ i \in C_j\}|\phi_j &\overset{\text{iid}}{\sim} f(\cdot|\phi_j) \text{ for } j = 1, \ldots, k \\ \phi_1, \ldots, \phi_k &\overset{\text{iid}}{\sim} P_0.\end{aligned} \tag{22}$$

Now let $m(x; \boldsymbol{X}_{C_j}) = \int_\Theta f(x; \phi_j) \mathcal{L}(d\phi_j|\boldsymbol{X}_{C_j})$ the predictive density (under the parametric model) of X_{new}, given the subvector $\boldsymbol{X}_{C_j}$ of data in cluster j. If $f(\cdot; \phi)$ is conjugate w.r.t. P_0, the functions $m(x; \boldsymbol{X}_{C_j})$ have closed-form analytic expressions. Hence,

$$\begin{aligned}&\frac{1}{dx}\mathbb{P}(X_{new} \in dx, Y^\epsilon_{new}(X_i) = 1|data) \\ &\quad = \int \Big\{ w_1(n_{l_1^h} - \sigma) m(x; \boldsymbol{x}_{C_1}) + \cdots + w_1(n_{l^h_{k_h^\epsilon}} - \sigma) m(x; \boldsymbol{x}_{C_{l^h_{k_h^\epsilon}}}) \Big\} \mathcal{L}(d\boldsymbol{\rho}|data) + A,\end{aligned}$$

where $A := \int w_0 f(x;\theta_{new}) \mathbf{1}_{\{Y^{\epsilon}_{new}(X_i)=1\}} P_0(d\theta_{new}) \mathcal{L}(d\boldsymbol{\rho}, d\boldsymbol{\phi}|data)$. If we evaluate (18) for a given $X_i = x_i$ through a MCMC sample $\{\boldsymbol{\rho}^{(1)}, \ldots, \boldsymbol{\rho}^{(G)}\}$, $\{\boldsymbol{\phi}^{(1)}, \ldots, \boldsymbol{\phi}^{(G)}\}$ from $\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\phi}|data)$, then a MCMC estimate of (18) is

$$\frac{1}{G}\sum_{g=1}^{G}\left\{ w_1(n^{(g)}_{l^h_1} - \sigma) m(x; \boldsymbol{x}_{C^g_1}) + \cdots + w_1(n^{(g)}_{l^h_{k^{\epsilon}_h}} - \sigma) m(x; \boldsymbol{x}^{(g)}_{C_{l^h_{k^{\epsilon}_h}}}) + A^{(g)} \right\}.$$

In order to compute the integral $A^{(g)}$, we will resort to an importance sampling algorithm, with importance function

$$w_1(n^{(g)}_{l^h_1} - \sigma)\pi(d\theta_{new}|C^{(g)}_1) + \cdots + w_1(n^{(g)}_{l^h_{k^{\epsilon}_h}} - \sigma)\pi(d\theta_{new}|C^{(g)}_{l^h_{k^{\epsilon}_h}}), \tag{23}$$

where $\pi(d\theta_{new}|C^{(g)}_1) \propto \prod_{i\in C_j} f(x_i;\phi_j) P_0(d\phi_j)$, with $j \in \{l^h_1, \ldots, l^h_{k^{\epsilon}_h}\}$, are the posterior distributions of θ_{new} under the parametric model (22) defined above.

As far as the denominator of (20) is concerned, it is the posterior predictive distribution of a new observation under (3). We will compute it from a MCMC sample from $\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\phi}|data)$, as usually done in the Bayesian context.

On the other hand, the denominator in (19) is the integral, w.r.t. X_{new}, of the numerator; therefore, for $i = 1, \ldots, n$, we have

$$\begin{aligned} K(X_i) &:= \int_{\mathbb{R}^p} \mathbb{P}(X_{new} \in dx, Y^{\epsilon}_{new}(X_i) = 1|data) \\ &= \int w_1(n_{l^h_1} - \sigma) + \cdots + w_1(n_{l^h_{k^{\epsilon}_h}} - \sigma)\mathcal{L}(\boldsymbol{\rho}|data) + B, \end{aligned}$$

where $\{l^h_1, \ldots, l^h_{k^{\epsilon}_h}\}$ are such that $i \in C^{\epsilon}_h = C_{l^h_1} \cup \cdots \cup C_{l^h_{k^{\epsilon}_h}}$, and

$$B := \int w_0 \mathbb{I}_{\{Y^{\epsilon}_{new}(X_i)=1\}} P_0(d\theta_{new}) \mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\phi}|data).$$

Clearly, a MCMC estimation of $K(X_i)$ is:

$$\hat{K}(X_i) = \frac{1}{G}\sum_{g=1}^{G}\left\{ w_1(n^{(g)}_{l^h_1} - \sigma) + \cdots + w_1(n^{(g)}_{l^h_{k^{\epsilon}_h}} - \sigma) + B^{(g)} \right\}.$$

Similarly as before, in order to compute the integral $B^{(g)}$, we will resort to an importance sampling algorithm, with importance importance function defined in (23).

Finally to compute (16), for a fixed i, we need to evaluate

$$\begin{aligned} &\mathbb{E}(X_{new}|Y^{\epsilon}_{new}(X_i) = 1, data) = \int_{\mathbb{R}^p} x\mathbb{P}(X_{new} \in dx|Y^{\epsilon}_{new} = 1, data) \\ &= \frac{1}{K(X_i)} \int_{\mathbb{R}^p} x\mathbb{P}(X_{new} \in dx, Y^{\epsilon}_{new} = 1|data) \\ &= \frac{1}{K(X_i)} \int w_1(n_{l^h_1} - \sigma)\mathbb{E}(X_{new}|\boldsymbol{X}_{C_{l_1}}) + \cdots + w_1(n_{l^h_{k^{\epsilon}_h}} - \sigma)\mathbb{E}(X_{new}|\boldsymbol{X}_{C_{l^h_{k^{\epsilon}_h}}})\mathcal{L}(\boldsymbol{\rho}|data) + C, \end{aligned}$$

where with $\mathbb{E}(X_{new}|\boldsymbol{X}_{C_{l_j}})$, $j = l_1^h, \dots, l_{k_\epsilon^h}^h$, we denote the predictive mean under the parametric model in (22), while

$$C := \int w_0 \mu(\theta_{new}) \mathbb{I}_{\{Y_{new}^\epsilon(X_i)=1\}} P_0(d\theta_{new}) \mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\phi}|data),$$

with $\mu(\theta_{new}) := \int_{\mathbb{R}^p} x f(x; \theta_{new}) dx$. Observe now that if P_0 and $f(\cdot, \phi)$ are conjugate, the posterior mean is easy to compute analytically, so that $\mathbb{E}(X_{new}|Y_{new}^\epsilon(X_i) = 1, data)$ can be estimated by

$$\frac{1}{\hat{K}(X_i)} \frac{1}{G} \sum_{g=1}^{G} \left\{ w_1(n_{l_1^h}^{(g)} - \sigma)\mathbb{E}(X_{new}|\boldsymbol{X}_{C_{l_1}}^{(g)}) + \dots + w_1(n_{l_{k_h^\epsilon}^h}^{(g)} - \sigma) + \mathbb{E}(X_{new}|\boldsymbol{X}_{C_{\substack{l^h \\ k_\epsilon^h}}}^{(g)}) + C^{(g)} \right\}.$$

The term $C^{(g)}$ can be evaluated using the importance function (23). Computations to estimate $\mathbb{E}(X_{new}^2|Y_{new}^\epsilon(X_i) = 1, data)$ and hence $\mathrm{Var}(X_{new}|Y_{new}^\epsilon(X_i) = 1, data)$ are very similar and we will skip here the details.

$\overset{\epsilon}{\leftrightarrow}$ *is an equivalence relation.*
Let $\mathcal{X} := \{X_1, X_2 \dots\}$ a sequence of data, and let $\{\theta_1, \theta_2, \dots\}$ be a sequence of labels attached to $\mathcal{X}$, such that, for each i, $X_i|\theta_i \sim f(\cdot|\theta)$. Let $\epsilon \geq 0$ and $d(\cdot, \cdot)$ a distance between densities; we prove that the relation $\overset{\epsilon}{\leftrightarrow}$, defined in Definition 1 and 2, is an equivalence relation on $\mathcal{X}$, i.e. it is reflexive, symmetric and transitive.
Reflexivity: Let $X_i \in \mathcal{X}$; then trivially $d(f(\cdot, \theta_i), f(\cdot, \theta_i)) = 0 \leq \epsilon$, so that then $X_i \overset{\epsilon}{\rightsquigarrow} X_i$ and hence $X_i \overset{\epsilon}{\leftrightarrow} X_i$.
Symmetry: Suppose that $X_i \overset{\epsilon}{\leftrightarrow} X_j$; then by Definition 2, there exist a finite sequence of index $\{j_1, \dots, j_{m_j}\}$ such that

$$X_i \overset{\epsilon}{\rightsquigarrow} X_{j_1} \overset{\epsilon}{\rightsquigarrow} X_{j_2} \overset{\epsilon}{\rightsquigarrow} \dots \overset{\epsilon}{\rightsquigarrow} X_{j_m} \overset{\epsilon}{\rightsquigarrow} X_j.$$

Hence, the sequence $\{j_m, \dots, j_1\}$ is such that

$$X_j \overset{\epsilon}{\rightsquigarrow} X_{j_m} \overset{\epsilon}{\rightsquigarrow} \dots \overset{\epsilon}{\rightsquigarrow} X_{j_2} \overset{\epsilon}{\rightsquigarrow} X_{j_1} \overset{\epsilon}{\rightsquigarrow} X_i,$$

so that $X_j \overset{\epsilon}{\leftrightarrow} X_i$.
Transitivity: If $X_i \overset{\epsilon}{\leftrightarrow} X_j$ and $X_j \overset{\epsilon}{\leftrightarrow} X_k$, then there exists two set of indexes $\{j_1, \dots, j_{m_j}\}$ and $\{k_1, \dots, k_{m_k}\}$, such that

$$X_i \overset{\epsilon}{\rightsquigarrow} X_{j_1} \overset{\epsilon}{\rightsquigarrow} X_{j_2} \overset{\epsilon}{\rightsquigarrow} \dots \overset{\epsilon}{\rightsquigarrow} X_{j_m} \overset{\epsilon}{\rightsquigarrow} X_j \overset{\epsilon}{\rightsquigarrow} X_{k_1} \overset{\epsilon}{\rightsquigarrow} \dots \overset{\epsilon}{\rightsquigarrow} X_{k_{m_k}} \overset{\epsilon}{\rightsquigarrow} X_k;$$

hence $X_i \overset{\epsilon}{\leftrightarrow} X_k$.

References

Antoniak, C. E. (1974). "Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems." *The Annals of Statistics*, 2, 1152–1174.

Argiento, R., Guglielmi, A., and Pievatolo, A. (2009). "A comparison of nonparametric priors in hierarchical mixture modelling for AFT regression." *Journal of Statistical Planning and Inference*, 139, 3989–4005.

— (2010). "Bayesian density estimation and model selection using nonparametric hierarchical mixtures." *Computational Statistics and Data Analysis*, 54, 816–832.

Argiento, R., Guglielmi, A., and Soriano, J. (2012). "A semiparametric Bayesian generalized linear mixed model for the reliability of Kevlar fibres." *Applied Stochastic Models in Business and Industry*, to appear.

Binder, D. A. (1978). "Bayesian Cluster Analysis." *Biometrika*, 65, 31–38.

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. (1998). "A genome-wide transcriptional analysis of the mitotic cell cycle." *Molecular Cell*, 2, 65–73.

Cremaschi, A. (2012). "Model-based clustering via Bayesian nonparametric mixture models." Tesi di laurea magistrale, Ingegneria Matematica, Politecnico di Milano.

Dahl, D. B. (2009). "Modal Clustering in a Class of Product Partition Models." *Bayesian Analysis*, 4, 631–652.

Ester, M., Kriegel, H. P., and Xu, X. (1996). "Knowledge Discovery in Large Spatial Databases: Focusing Techniques for Efficient Class Identification." In *Proc. 4th Int. Symp. on Large Spatial Databases, Portland, ME, 1995, Lecture Notes in Computer Science*, volume 951, 67–82. Springer.

Ewens, W. J. (1972). "The sampling theory of selectively neutral alleles." *Theoretical Population Biology*, 3, 87–112.

Ferguson, T. S. (1973). "A Bayesian analysis of some nonparametric problems." *The Annals of Statistics*, 1, 209–230.

Fritsch, A. and Ickstadt, K. (2009). "Improved Criteria for Clustering Based on the Posterior Similarity Matrix." *Bayesian Analysis*, 4, 367–392.

Gelfand, A. E., Dey, D. K., and Chang, H. (1992). "Model determination using predictive distributions, with implementation via sampling-based methods (with discussion)." In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M. (eds.), *Bayesian Statistics 4*, 147–167. New York: Oxford University Press.

Griffin, J. (2010). "Default priors for density estimation with mixture models." *Bayesian Analysis*, 5, 45–64.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). "On Clustering Validation Techniques." *Journal of Intelligent Information Systems*, 17, 107–145.

Hennig, C. (2012). *fpc: Flexible procedures for clustering*. R package version 2.1-4. URL `http://CRAN.R-project.org/package=fpc`

Hubert, L. J. and Arabie, P. (1985). "Comparing partitions." *Journal of Classification*, 2, 193–218.

Johnson, S. (1967). "Hierarchical clustering schemes." *Psychometrika*, 32, 241–254.

Lau, J. W. and Green, P. J. (2007). "Bayesian model based clustering procedures." *Journal of Computational and Graphical Statistics*, 16, 526–558.

Lee, J., Quintana, F., Müller, P., and Trippa, L. (2012). "Defining predictive probability functions for species sampling models." *Statistical Science*, Forthcoming.

Lijoi, A., Mena, R. H., and Prünster, I. (2007). "Controlling the reinforcement in Bayesian nonparametric mixture models." *Journal of the Royal Statistical Society B*, 69, 715–740.

MacQueen, J. (1967). "Some Methods for classification and Analysis of Multivariate Observations." In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, 281–297. University of California Press.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Hoboken, NJ (USA): Wiley.

Medvedovic, M., Yeung, K., and Bumgarner, R. (2004). "Bayesian mixture model based clustering of replicated microarray data." *Bioinformatics*, 20, 1222–1232.

Neal, R. (2000). "Markov Chain sampling Methods for Dirichlet process mixture models." *Journal of Computational and Graphical Statistics*, 9, 249–265.

Pitman, J. (1996). "Some Developments of the Blackwell-Macqueen urn Scheme." In Ferguson, T. S., Shapley, L. S., and B., M. J. (eds.), *Statistics, Probability and Game Theory: Papers in Honor of David Blackwell*, volume 30 of *IMS Lecture Notes-Monograph Series*, 245–267. Hayward (USA): Institute of Mathematical Statistics.

Qin, Z. S. (2006). "Clustering microarray gene expression data using weighted Chinese restaurant process." *Bioinformatics*, 22, 1988–1997.

Quintana, F. A. and Iglesias, P. L. (2003). "Bayesian Clustering and Product Partition Models." *Journal of the Royal Statistical Society B*, 65, 557–574.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. URL `http://www.R-project.org/`

Rand, W. M. (1971). "Objective Criteria for the Evaluation of Clustering Methods." *Journal of the American Statistical Association*, 66, 846–850.

Regazzini, E., Lijoi, A., and Prünster, I. (2003). "Distributional results for means of random measures with independent increments." *The Annals of Statistics*, 31, 560–585.

Rousseeuw, P. A. (1987). "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis." *Journal of Computational and Applied Mathematics*, 20, 53 –65.